THE INVESTIGATION OF COST VARIANCES MODELED AS A PARTIALLY OBSERVABLE MARKOV PROCESS

A THESIS

Presented to

The Faculty of the Division of Graduate Studies

By

Betty J. Largay

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Operations Research

Georgia Institute of Technology

July, 1978

THE INVESTIGATION OF COST VARIANCES MODELED

AS A PARTIALLY OBSERVABLE MARKOV PROCESS

Approved:

Donovan Young, Chairman

George E. Monahan

Thomas B. Clark

Date approved by Chairman 4 August 1976

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my advisor, Dr. Donovan Young, for his assistance and support which enabled the completion of this study. His constructive suggestions and editorial comments did much to shape the final form of this research. I also thank Dr. George Monahan for his guidance and invaluable assistance throughout and Dr. Thomas Clark whose service on the thesis committee is appreciated.

Mostly, I thank my husband, Jim, for his assistance in developing the various drafts and in his patience and constant encouragement during the time spent in the preparation of this thesis.

TABLE OF CONTENTS

TABLE OF CONTENTS (Contd)

LIST OF TABLES

LIST OF ILLUSTRATIONS

SUMMARY

Standard cost systems are widely used in cost accounting. Differences between reported costs and the corresponding standard costs are known as cost variances and are considered valuable, though incomplete, indicators of abnormal conditions.

In addition to the cost variance analysis literature treating decomposition of costs, control charts and elementary statistical procedures, there is a literature on cost-variance investigation treating the problem of inferring amounts of managerial investigation economically justified by cost variances of various sizes. This literature declined abruptly after 1969, when papers by R.S. Kaplan and T.R. Dyckman established that the appropriate model was that of the partially observable Markov process, a model which at that time presented insurmountable computational burdens for problems of practical size.

Partially observable Markov processes also arise in engineering contexts, and computational progress stemming from Edward Sondik's "One-pass Algorithm" has been striking enough to warrant reconsideration of implementing these models in cost accounting systems.

In this thesis, the One-pass Algorithm is applied to formulation and solution of cost variance investigation problems.

Computability and efficiency of the algorithm are demonstrated by applying it to the models suggested by Kaplan and Dyckman, and by applying it to extensions of these models that allow less severe data requirements at a cost of additional computational load.

An extension of Dyckman's procedure for estimating Markov transition probabilities is presented. Dyckman's procedure is a modification of traditional estimation methods, to allow for using a partially observable historical record, but applies to systems with only two states. The extension allows arbitrary numbers of states.

A heuristic modification is presented which eliminates some of the calculations necessary in the One-pass Algorithm. The modification is found to contribute to computational efficiency.

Finally, a fully documented Fortran computer program is presented for the solution of partially observable Markov processes by the One-pass Algorithm. It includes both the finite-horizon and infinite-horizon versions of the algorithm, it allows arbitrary numbers of states and decisions, and it is written so as not to be restricted to the cost-accounting context. It is believed to be the first implementation of the algorithm to be made available in the open literature.

CHAPTER I

INTRODUCTION

In management, a widely used tool is the standard cost system wherein a standard or budgeted amount is specified for the next period for each of the processes in the system. A cost variance occurs when an actual cost differs from the corresponding standard cost.

Very small cost variances are ignored. Very large ones are investigated. An investigation will at least include decomposition of costs into components such as direct labor, direct materials, factory overhead, etc., to identify the specific abnormality so as to allow effective managerial intervention if needed. In the conventional two-factor analysis, for example, the total cost variance is subdivided under the assumption that the cost is a product of two factors--a unit price or cost, and a quantity or usage [1,5,9]. At its most elaborate, a cost variance investigation can go far beyond these routine techniques and involve studies consuming months of effort.

The absolute or proportional size of a cost variance, or even its statistically normalized size, can provide only a rough measure of its true significance to management, since this significance ultimately depends on the prospective net benefit of managerial action taken in response to the variance. A large or surprising variance is not necessarily significant unless it indicates a situation that is profitably controllable by management.

One responsibility of management, the one addressed in this thesis, is the responsibility to make control decisions based on the assessed significance of cost variances that are neither small enough to ignore nor large enough to be certain that action is called for.

The usual nature of the control decision is, first, whether to investigate the cause, and then whether to take corrective action. Kaplan [14] provides a complete survey of cost-variance investigation literature up to mid-1975.

Several investigators have dealt directly with the problem of obtaining control decisions [2,6,7,8,13,17,18]. In general, they have considered the problem to be one in which a process can be in one of two states, in-control or out-of-control. A process is said to be in control when a cost variance, if present, is caused by a factor not controllable by management; otherwise, it is said to be out of control. However, the actual state of the system is seldom known with certainty and must be inferred from the reported costs. Each of the several cost-variance investigation models is a procedure which uses the available knowledge concerning the state of the system to determine investigative policies.

In particular, several authors [7,8,13] have presented the problem as a partially observable Markov process, where the decision to investigate is a function of the probability that the process is operating in control, the cost of an investigation, and the cost of allowing an out-of-control situation to continue.

In the theory of Markov processes, a discrete-time, finite-state, stationary Markov process, for which the state cannot be determined with certainty is a partially observable Markov process [21,22,23,25,26,27]. A

state-related variable is observed, and the probabilities of the underlying states are estimated after each observation using Bayes' equation. In cost-variance investigations, the observed variable is the end-of-period reported cost, for which a known conditional probability distribution is assumed to exist.

Unfortunately, a partially observable Markov process implies that the underlying discrete state space becomes continuous due to the uncertainty about which state the system is in. This continuous representation of the discrete state space consists of probabilities of state occurrence, and it must be discretized in some manner to allow computation. A promising approach has been that of Sondik [26], whose method avoids arbitrary discretizing of the state space. By computing boundaries within which a given decision is optimal, Sondik's method obtains a finite set of regions equivalent to individual states for computational purposes.

This thesis has the following objectives:

1. Formulate the cost-variance investigation decision problem, for both a finite and infinite planning horizon, as a partially observable Markov process.

2. Apply Sondik's algorithm to solve the problem using simple finite and infinite horizon examples.

3. Write a documented Fortran program employing the algorithm developed in (2) above to solve the sample problems.

4. Demonstrate implementation of the algorithm by developing a test problem using hypothetical data and parameter values, with characteristics similar to those likely to be encountered in an actual decision situation.

5. Unify cost-variance investigation models in the accounting literature and extend these models (for example, to consider additional states beyond in-control and out-of-control) where the capabilities of Sondik's algorithm make extensions feasible.

6. Derive a procedure for estimating the probabilities in an N-state transition matrix.

The remainder of this thesis is organized along the following general lines. Chapter II is a brief survey of the major articles in accounting dealing with cost-variance investigation. In Chapter III the basic cost-variance investigation problem over a finite horizon is described, the notation is defined within the framework of a partially observable Markov process, and Sondik's algorithm is presented. The cost-variance investigation problem over an infinite horizon is treated in Chapter IV. Solutions of finite and infinite horizon problems using Sondik's algorithm are illustrated in Chapter V with some simple numerical examples. This chapter also includes discussion of the test problem and some extensions to the basic model. Chapter VI presents conclusions and indicates additional areas which would be worthy of further research.

CHAPTER II

OVERVIEW OF METHODS

The traditional accounting control model using standard costing is simple. The decision to investigate a variance, favorable or unfavorable, is solely determined by the absolute or relative size of the cost variance. For an example, see Juers [12]. The determination of standards and decision criteria are based on the experience and judgement of management. The strength of this model lies in its ease of application, but it ignores objective information that can be obtained from past observations; there is no explicit consideration of the cost of investigation or of the cost of failing to detect an out-of-control situation.

The classical Shewhart model is the first attempt at control based on classical statistics [24]. It is assumed that distinct probability distributions generate the reported costs for the system, conditional on the state of the system. It is customary to assume that the reported costs are normally distributed. The parameters (mean and standard deviation) of the in-control probability distribution are estimated from past observations when it was known that the system was operating in control. It is assumed that as long as the system continues operating in control that observations will conform to this probability distribution. When the process moves to an out-of-control state, the parameters are assumed to change.

For normally distributed cost observations, Shewhart's test is

designed to indicate a shift in the mean value of the distribution, μ. With this procedure, an investigation is called for whenever the current cost observation falls outside a predetermined statistical range of values, $\mu \pm n\sigma$, where n is a predetermined value and σ is the standard deviation when in control. Shewhart suggested using a 3σ limit for most applications and justified it empirically, finding that it results in a reasonable balance between unnecessary investigations on the one hand, and undiscovered trouble on the other. This procedure is equivalent to investigating the worst 0.135 percent of cost variances encountered. As with the traditional procedure, the decision to investigate considers only the current cost observation and fails to incorporate the cost of investigation or the cost of failing to correct an out-of-control situation.

There are a number of variations to the Shewhart control chart procedure which attempt to detect more quickly any shifts that have occurred in the probability distribution. For a summary of these variations, see Roberts [20]. An example is one where investigative action is called for when a run of observations falls outside the 1σ or 2σ level. Another example is the moving average test. An average of the last k observations is established and investigative action is called for if the average falls outside $\mu \pm L_k\sigma/k$, where L_k is a predetermined value.

The idea of incorporating rewards and costs associated with an investigation into a control decision is the major contribution of an article by Bierman, Fouraker and Jaedicke (BFJ) [2]. The BFJ model assumes that management can estimate the probability q of being in control for every given cost observation, θ. Given that the system may be in one of

two states and that there are two courses of action available in each state, a two-dimensional cost structure is established. In Table 1, C is the cost of an investigation and L the estimated present value of the cost savings from correcting an out-of-control situation.

Table 1. Cost Structure in the BFJ Model

Action / State	Investigate: a_1	Do Not Investigate: a_2
In-control: X_1	C	0
Out-of-control: X_2	C	L

The expected cost of each action is as follows:

$$E(a_1) = Cq + C(1-q) = C$$

$$E(a_2) = 0q + L(1-q) = L(1-q) \quad .$$

Therefore if $C < L(1-q)$, or $q < (L-C)/L$, an investigation is warranted. If $C \geq L(1-q)$, an investigation should not be made.

The major difficulty with this model lies in trying to estimate the value of L, which depends on how often the same situation may be expected to recur and how permanent an effect the management action may be expected to have.

In all these models, knowledge of the state of the process is

based solely on the current cost observation. The following Bayesian models are attempts to overcome this shortcoming by combining current and prior cost observations to develop state probabilities.

Dyckman [7] proposes a model with the same cost structure as the BFJ model, but extended in that it uses Bayesian updating of state probabilities, so that all information available since the last investigation is incorporated into the knowledge of the true state. It is assumed in Dyckman's model that the true state of the process undergoes transitions according to the following matrix P,

$$P = \begin{bmatrix} g & 1-g \\ 0 & 1 \end{bmatrix}$$

so that an in-control process has a 1-g probability of going out-of-control each time period, subsequently remaining out of control until the process is reset by management action. Dyckman indicates that the estimate of g is obtained by estimating the mean number of periods (passage time) before going out-of-control, which equals 1/(1-g) for this process. See Appendix A for a more complete discussion of estimating the parameters in the transition matrix.

Dyckman's model assumes that the probability density function of cost, $f_i(\theta)$, for each state i is known. See Figure 1. Because the observed cost is determined by a large number of additive factors, the cost observations will often be assumed to be normally distributed, but any known distribution can be used in the model.

In addition, the model makes the following assumptions:

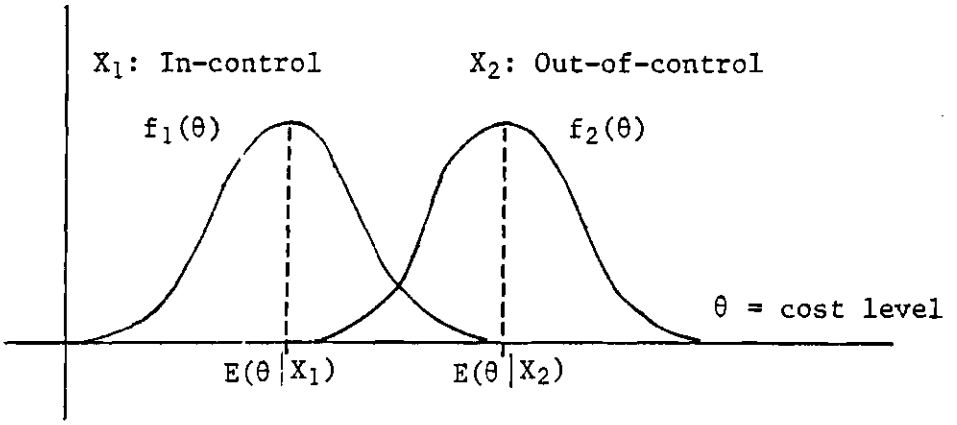


Figure 1. Cost Density Functions in Dyckman's Model

1. Stationarity: The transition probability matrix remains the same each period; that is, each entry is a function only of the state the system now occupies and the state it will occupy during the next time interval. Also, the cost of investigation is a constant, or its expected value is constant.

2. Beginning-of-period transitions: When the system makes a transition from one state to another, it does so at the beginning of the period or sufficiently early in the reporting period to effect the outcome of the period. Also, once the decision to investigate is made, the investigation and any necessary correction is assumed to take place immediately and to be fully effective.

At the beginning of period n, the state probabilities are given by $P_n(X_1)$ and $P_n(X_2) = 1 - P_n(X_1)$, where X_1 represents the in-control state and X_2 the out-of-control state. Assume that these probabilities are such that at the beginning of period n, no investigation is called for; that is, $P_n(X_1) \geq (L-C)/L$.

During the period a transition to another state may or may not occur. At the end of the period, a cost report is received. However, there is no assurance as to the state of the system, because a system operating in control can report a somewhat high cost or a system operating out of control can report a relatively low cost. As an example, a high cost observation for period n will increase the probability that the system will be out of control at the beginning of period n+1. The state probabilities for period n+1 are found through Bayesian updating:

$$P_{n+1}(X_1) = P_{n+1}(X_1|\theta) = \frac{gf_1(\theta)P_n(X_1)}{gf_1(\theta)P_n(X_1)+f_2(\theta)[1-gP_n(X_1)]}$$

$$P_{n+1}(X_2) = P_{n+1}(X_2|\theta) = 1 - P_{n+1}(X_1)$$

$$= \frac{f_2(\theta)[1-gP_n(X_1)]}{gf_1(\theta)P_n(X_1)+f_2(\theta)[1-gP_n(X_1)]}$$

An investigation is warranted if $P_{n+1}(X_1) < (L-C)/L$; otherwise, if $P_{n+1}(X_1) \geq (L-C)/L$, no action is taken.

Dyckman explores some variations in the assumptions of this model, some of which will be discussed later in this thesis.

Another model very similar to Dyckman's except with respect to the cost structure is formulated by Kaplan [13]. Instead of summarizing all cost information about the future in one parameter L, Kaplan's model uses dynamic programming to compute optimal policies that minimize discounted future costs. The assumptions of this two-state model are the same as

Dyckman's, except that end-of-period transitions are assumed, with cost observations obtained prior to transitions. The updated probabilities are:

$$P_{n+1}(X_1) = P_{n+1}(X_1|\theta) = \frac{gf_1(\theta)P_n(X_1)}{f_1(\theta)P_n(X_1)+f_2(\theta)P_n(X_2)}$$

$$P_{n+1}(X_2) = P_{n+1}(X_2|\theta) = 1 - P_{n+1}(X_1)$$

$$= \frac{(1-g)f_1(\theta)P_n(X_1)+f_2(\theta)P_n(X_2)}{f_1(\theta)P_n(X_1)+f_2(\theta)P_n(X_2)}$$

In establishing the recursive equations necessary for the dynamic programming solution to the problem, Kaplan first defines

$$T(q|\theta) = \frac{gf_1(\theta)q}{f_1(\theta)q+f_2(\theta)(1-q)}$$

where q is the probability of presently being in control and T is the operator which updates the probability q after receiving cost observation θ. Let $C_n(q)$ be the minimum expected cost that can be achieved given that the system initially has probability q of being in control and that an optimal policy is followed for the next n periods. Since there are two courses of action, there are two possible values for the expected cost. In the first case, where there is an investigation with cost K, the expected one-period cost is:

$$K + \sum_{\theta} [\theta\{gf_1(\theta) + (1-g)f_2(\theta)\}]$$

where $[g\ f_1(\theta) + (1-g)\ f_2(\theta)]$ is the unconditional probability density function for θ.

In addition to the expected one-period cost, the expected future cost for the remaining $n-1$ periods is given by:

$$\beta \sum_{\theta} C_{n-1}\big(T(g|\theta)\big)\,[gf_1(\theta) + (1-g)f_2(\theta)]$$

where $0 < \beta \leq 1$ is the single-period present-worth discount factor. The value $C_{n-1}\big(T(g|\theta)\big)$ represents the minimum expected cost when there are $n-1$ periods until termination, and the updated probability of being in control is $T(g|\theta)$. Therefore the total expected cost incurred if an investigation is undertaken is

$$K + \sum_{\theta} [\theta + \beta C_{n-1}\big(T(g|\theta)\big)]\big(gf_1(\theta) + (1-g)f_2(\theta)\big)$$

If the action chosen is to do nothing, then the expected one-period cost is given by

$$\sum_{\theta} [\theta\big(qf_1(\theta) + (1-q)f_2(\theta)\big)]$$

and expected future cost is

$$\beta \sum_{\theta} C_{n-1}\big(T(q|\theta)\big)\,[qf_1(\theta) + (1-q)f_2(\theta)]$$

Thus the optimal total expected cost is

$$C_n(q) = \min \begin{cases} K + \sum_{\theta} [\theta + \beta C_{n-1}(T(g|\theta))](gf_1(\theta) + (1-g)f_2(\theta)) \\ \\ \sum_{\theta} [\theta + \beta C_{n-1}(T(q|\theta))](qf_1(\theta) + (1-q)f_2(\theta)) \end{cases}$$

and the optimal policy is dictated accordingly. For an infinite planning horizon, the optimal policy is found from the same set of recursive equations by letting n increase until the state probabilities approach limiting values.

The assumption of having only two states for any given process is a feature of nearly all the formal models for investigating cost variances, although, in practice, processes may tend to move gradually away from a state of being in control rather than suddenly jumping to a single out-of-control situation. An approach which relaxes the two-state assumption is suggested by Duvall [6]. He considers a process having a continuum of states where the state of the process is the level of the controllable part of a cost variance. Every cost variance x is made up of two components--a non-controllable component w arising randomly and a component y generated by off-standard performance. The assumptions of the model are as follows:

1. The probability distribution of the non-controllable part of the cost variance is normal with $\mu_w = 0$ and standard deviation σ_w, determined from past data collected while in control.

2. The probability distribution of the controllable part of the cost variance is normal with mean μ_y and standard deviation σ_y.

3. The contribution of controllable and non-controllable parts of the cost variance are independent. Therefore the estimate of the mean of

the controllable part of the cost variance is the mean value μ_x of observed cost variances over several periods of time.

$$\mu_y = \mu_x$$

The relationship of the standard deviations can be expressed as

$$\sigma_x^2 = \sigma_y^2 + \sigma_w^2$$

The correlation of the observed cost variance and the cost variance due to out-of-control performance is

$$\rho = \sigma_y^2 / \sigma_x \sigma_y$$

$$= \sigma_y / \sigma_x$$

When a cost variance is observed, the distribution of the variance due to out-of-control performance is revised according to the formula provided by Duvall.

$$\mu_{y|x} = \mu_y + \rho\sigma_y(x-\mu_x)/\sigma_x$$

$$= \mu_y(1-\rho^2) + \rho^2 x$$

and

$$\sigma_{y|x} = \sigma_y \sqrt{1 - \rho^2}$$

In order to determine if an investigation is warranted, the costs

and benefits must be known. As with Dyckman, Duvall summarizes future cost savings in a single parameter for each state. He assumes this relationship to be linear;

$$L(y) = \begin{cases} k_1 y - C & \text{if } y > 0 \\ -(k_2 y + C & \text{if } y \leq 0 \end{cases}$$

where C is the cost of an investigation, k_1 is the present value of additional future profits per dollar of favorable cost variance not exploited, and k_2 is the present value of additional future cost per dollar of unfavorable cost variance not corrected. The expected profit V can be determined from the equation

$$E(V) = \int_{-\infty}^{0} \left(k_2 y + C\right)/\sqrt{2\pi}\,\sigma_{y|x}\Big) \exp\left(-(y-\mu_{y|x})^2/2\sigma^2_{y|x}\right) dy$$

$$+ \int_{0}^{\infty} \left(k_1 y - C\right)/\sqrt{2\pi}\,\sigma_{y|x}\Big) \exp\left(-(y-\mu_{y|x})^2/2\sigma^2_{y|x}\right) dy$$

If E(V) is positive, an investigation would take place; otherwise, no action would be taken.

The work reviewed in this chapter covers the major techniques currently available to determine when cost variances arising in a standard cost system should be investigated. The basic assumptions and some of the limitations of these procedures have been identified.

CHAPTER III

COST VARIANCE INVESTIGATION DECISIONS AS A PARTIALLY OBSERVABLE MARKOV PROCESS (POM)

This chapter extends the two-state Dyckman and Kaplan cost-variance investigation models [7,13] to a finite number of states, $1,2,\ldots,N_s$, where the states are assumed ordered according to desirability with state 1 most desirable. The new model parallels Dyckman's with respect to the basic assumptions but treats future costs within a dynamic programming formulation. The departures from Kaplan come mainly in the ordering of events and the redefinition of the state space.

This model has the following properties:

1. Transitions between states are in accordance with a stationary Markov process and occur at the beginning of the period.

2. The cost of investigation and correction remains constant from period to period. Let K_i be the cost incurred if the process is in state i. It is assumed that an investigation and possible correction take place immediately and are fully effective, thus returning the system to state 1.

3. The process is characterized by a finite operating time.

4. The action set consists of two elements, $A = \{a_1, a_2\}$, where a_1 is investigate and a_2 is do nothing.

5. Cost observations, received at the end of the period, are samples from discrete, state-related cost distributions. It is assumed that a reported cost, θ, is from a finite set of M possible cost observations and that elements in the set are not necessarily one-to-one with the states.

Later in the thesis, items (3) and (4) in the list above will be modified. Specifically, an infinite planning horizon will be considered in Chapter IV and an expanded action set will be examined in Chapter V.

Details of the POM Model

At the beginning of period n, management must make a control decision based on the cost report received at the end of the previous period along with all past knowledge of the process, but without necessarily full knowledge about the current state of the system. A control alternative, $a \in A$, is selected so as to minimize discounted expected total cost over the entire planning horizon. After a decision is made and implemented, the process undergoes a transition represented by the $N_s \times N_s$ matrix P^a.

$$P^a = \begin{bmatrix} p^a_{11} & p^a_{12} & \cdots & p^a_{1,N_s} \\ \vdots & \vdots & & \\ p^a_{N_s,1} & p^a_{N_s,1} & \cdots & p^a_{N_s,N_s} \end{bmatrix},$$

where p^a_{ij} is the probability that the process makes the transition from state i to state j, given that alternative $a \in A$ was selected.

At the end of the period, a cost report is received where the probability of a particular cost being reported is given by the stochastic probability cost matrix, R.

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1,M} \\ \vdots & & & \\ r_{N_s,1} & r_{N_s,1} & \cdots & r_{N_s,M} \end{bmatrix}.$$

Each element of R, $r_{j\theta}$, is the probability that the reported cost is θ, given that the true state of the process is j. Note that the set of possible cost observations is independent of the alternative selected, a property not necessarily belonging to a partially observable Markov process in general.

Since many of the cost values for one state are usually also possible for one or more other states, the true state of the process is not known exactly. If there existed a perfect correspondence between states of the process and cost observations, then the true state would immediately become known. The problem would reduce to a completely observable Markov process with its solution determined using Howard's policy iteration algorithm [10]. Since this is not the case, the decision made each period is conditioned on the probabilities that the process is found to be in each of the states rather than on the states of the process themselves. These probabilities are updated each period by Bayesian methods so as to include the effects of both the Markovian behavior of the process as well as the information available through the cost report. This is an example of a partially observable Markov process, as described by [22,26].

The underlying discrete-time, finite-state Markov process is modeled as a discrete-time, continuous-state Markov process. The state is the

probability vector $\pi = [\pi_1, \pi_2, \ldots, \pi_{N_s}]$, where π_i is the probability that the current state is i. The value π is said to be the state of knowledge of the underlying process.

For example, in a three-state process the state of knowledge may be represented as a point in the simplex shown in Figure 2. The i-th coordinate of the point indicates the perpendicular distance from the point to the side opposite the i-th vertex.

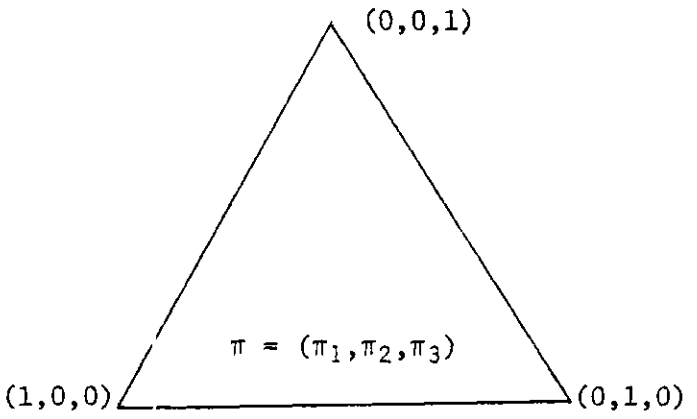


Figure 2. The Set Π for a Three-state Process

Given that the probability vector at the beginning of period n is π, and a cost of θ is observed after implementing alternative $a \in A$, the updated probability vector at the beginning of period n+1 is $\pi' = [\pi'_1, \pi'_2, \ldots, \pi'_n]$ where

$$\pi'_j = \frac{\sum_i \pi_i P^a_{ij} r_{j\theta}}{\sum_i \sum_j \pi_i P^a_{ij} r_{j\theta}} = \frac{\sum_i \pi_i P^a_{ij} r_{j\theta}}{Pr\{\theta | \pi, a\}} \qquad (1)$$

For simplicity, define T as the operator which updates π by equation (1), given θ and $a \epsilon A$.

$$\pi' = T(\pi | \theta, a)$$

In matrix notation,

$$\pi' = \frac{\pi P^a R_\theta}{P\{\theta | \pi, a\}}$$

where R_θ is an $N_s \times N_s$ diagonal matrix, the values on the diagonal being the values in the θ-column of R.

In determining the optimal policy it is necessary to first calculate the immediate expected cost for the process for each available decision. When the decision chosen is to investigate, a cost of K_i will be incurred if the process is in state i. K_i may be thought of as an expected cost of investigation and correction for state i.

Since an investigation guarantees a return to state 1, the transition matrix under this alternative has rows which are all identical. Each of these rows is the same as the first row in the transition matrix for the do-nothing alternative. Thus the immediate expected cost of investigation when in state i is given by

$$\gamma_i^{a_1} = K_i + \sum_{j,\theta} P_{ij}^{a_1} r_{j\theta} \cdot \theta$$

where a_1 is the decision to investigate.

The immediate expected cost when in state i for a do-nothing

decision is given by

$$\gamma_i^{a_2} = \sum_{j,\theta} P_{ij}^{a_2} r_{j\theta} : \theta$$

where a_2 indicates that no action is chosen.

In general, γ^a is the vector of immediate expected costs under alternative $a \epsilon A$. If π is the current probability vector, then the immediate expected cost is found by $\pi\gamma^a$.

Define $C^n(\pi)$ as the minimum discounted expected total cost given that: (1) there are n periods remaining until termination of the process, (2) the state probability vector at the beginning of the current period is π, and (3) an optimal policy is followed for the next n periods. It can be shown that $C^n(\pi)$ satisfies the recursion:

$$C^n(\pi) = \min_a \{ \sum_i \pi_i \gamma_i^a + \beta \sum_{i,j} \pi_i P_{ij}^a \sum_\theta r_{j\theta} C^{n-1}(T(\pi|\theta,a)) \}$$

which is equivalent to

$$C^n(\pi) = \min \{ \pi\gamma^a + \beta \sum_\theta P\{\theta|\pi,a\} C^{n-1}(T(\pi|\theta,a)) \} \qquad (2)$$

The One-pass Algorithm

The recursive equation (2) must be solved in order to determine optimal policies regarding the investigation of cost variances. For two-state problems, equation (2) can be expressed in one unknown, since the probabilities of the states sum to one, and its solution is relatively straightforward using backward recursion. In problems with greater than

two states, finding the solution to (2) is more difficult. One method is to divide the probability state space into a grid and solve (2) by conventional methods, such as Howard's policy iteration algorithm. Unfortunately, this suffers from the "curse of dimensionality" in that the amount of computational work increases exponentially as the underlying state space increases.

Sondik [26] has developed a technique, called the One-pass Algorithm, which is used to solve partially observable Markov processes. The algorithm enables a significantly larger size problem to be solved than by the usually encountered procedure of discretizing the probability state space by means of a regular grid. With the One-pass Algorithm, as the underlying state space, N_s, increases, the amount of computational work increases at a rate proportional to $(N_s)^I$ where I is between two and four.

Properties of the Cost Function

In situations where there are only a finite number of periods until termination, there exists an important property of the function $C^n(\cdot)$ which allows its solution to be expressed in a simple form. This property states that over the set Π, $C^n(\cdot)$ is piecewise linear and concave [26]. See Figure 3. The piecewise linearity causes the probability state space Π to be partitioned into a finite set of regions $\{R_1^n, R_2^n, \ldots, R_{S(n)}^n\}$.‡ Due to this property, $C^n(\pi)$ can be rewritten as

$$C^n(\pi) = \min_k \{\pi \alpha_k^n\} \quad k=1,\ldots,S^{(n)} \tag{3}$$

‡ Whenever the superscript n appears on a variable, n is the number of periods until termination.

where α_k^n is an α-vector from the set $\{\alpha_1^n, \alpha_2^n, \ldots, \alpha_{S(n)}^n\}$. Each α-vector corresponds to one of the R_k^n regions of the probability state space and can be thought of as the slope of $C^n(\pi)$ at π. Associated with each of the α-vectors is a decision alternative which is the optimal decision for all values of π that lie in that region.

For example, in Figure 3, assume that π lies in region R_4^n. Associated with R_4^n is α-vector α_4^n which gives

$$C^n(\pi) = \pi \alpha_4^n$$

The decision $a = \delta_4^n$, associated with α_4^n is the optimal decision for $\pi \in R_4^n$.

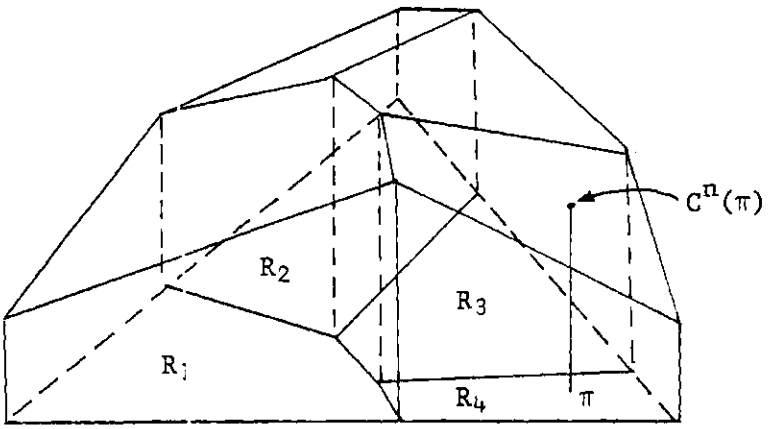


Figure 3. Cost Function for a Three-state Process

The α-vectors and their associated decision rules are determined by applying Sondik's One-pass Algorithm for partially observable Markov processes. Once the set of α-vectors has been found, to determine the optimal decision for a given π, find α_k^n such that

$$C^n(\pi) = \min_j \pi \alpha_j^n \qquad j=1,\ldots,S^{(n)}$$

The decision alternative, δ_k^n, associated with α_k^n is the optimal decision.

To see how these α-vectors are calculated, recall equation (2). For a specific $\pi \in R_k^n$

$$C^n(\pi) = \min_a \{\pi r^a + \beta \sum_\theta Pr\{\theta|\pi,a\} C^{n-1}(T(\pi|\theta,a))\} \tag{2}$$

From (3),

$$C^{n-1}(T(\pi|\theta,a)) = \min_j T(\pi|\theta,a)\, \alpha_j^{n-1} \; ; \; j=1,\ldots,S^{(n-1)}$$

Substituting this into (2),

$$C^n(\pi) = \min_a \{\pi r^a + \beta \sum_\theta \min_j \{\theta|\pi,a\} T(\pi|\theta,a) \alpha_j^{n-1}\} \quad ,$$

$$j=1,\ldots,S^{(n-1)}$$

Since

$$T(\pi|\theta,a) = \frac{\pi P^a R_\theta}{Pr\{\theta|\pi,a\}}$$

then

$$C^n(\pi) = \min_a \{\pi \alpha^a + \beta \sum_\theta \min_j \pi P^a R_\theta \alpha_j^{n-1}\} \tag{4}$$

$$j=1,\ldots,S^{(n-1)}$$

Let a* be the minimizing alternative for (4). Then for $\pi \in R_k^n$,

$$\alpha_k^n = \gamma^{a*} + \beta \sum_{\theta} \min_{j} P^{a*} R_\theta \alpha_j^{n-1} \quad ; \quad j=1,\ldots,S^{(n-1)} \tag{5}$$

Therefore, given the set of α-vectors for $n-1$ remaining time periods, the entire set of α-vectors for n remaining periods can be calculated using (4) and (5).

However, an α-vector cannot be explicitly calculated for every point in the set Π, since Π is an infinite set. One possible method for calculating the α-vectors from the given information is to divide the probability state space into a regular grid. The α-vectors from this finite set of π-values can be calculated and a list made of different α-vectors. There is, however, no guarantee that all of the α-vectors for a particular time period will be found.

In the One-pass Algorithm, the partitioning of the probability state space induced by the piecewise linearity of the cost function is used directly in calculating the α-vectors. In each region of the partition, $C^n(\pi)$ is a linear function of π, and this finite set of regions forms a one-to-one correspondence with the set of α-vectors in the final solution.

The Initial Step of the Algorithm

Assume that the number of remaining time periods is $n \geq 1$ and that the set $\{\alpha_j^{n-1},\ldots,\alpha_{S^{(n-1)}}^{n-1}\}$ has been determined. For $n = 1$, the set $\{\alpha_j^{n-1}\}$ is the terminal cost vector. For each $\pi \in \Pi$, the mapping function $d(\pi,\theta,a)$ is defined as the minimizing subscript in the equation

$$\pi P^{a} R_{\theta} \alpha^{n-1}_{d(\pi,\theta,a)} = \min_{j} \pi P^{a} R_{\theta} \alpha^{n-1}_{j} \quad ,$$

$$j-1,\ldots,S^{(n-1)}$$

That is, for a given π, $d(\pi,\theta,a)$ maps the pair (θ,a) onto the set of the α^{n-1}-vectors. Thus $C^{n}(\pi)$ can be written as

$$C^{n}(\pi) = \min_{a}\{\pi \gamma^{a} + \beta \sum_{\theta} \pi P^{a} R_{\theta} \alpha^{n-1}_{d(\pi,\theta,a)}\}$$

To begin the algorithm, choose an initial π_0, say $\pi_0 = (1,0,\ldots,0)$. For π_0, determine the mapping for each pair (θ,a) and calculate $C^{n}(\pi_0)$.

$$C^{n}(\pi_0) = \min_{a}\{\pi_0 \gamma^{a} + \beta \sum_{\theta} \pi_0 P^{a} R_{\theta} \alpha^{n-1}_{d(\pi_0,\theta,a)}\} \tag{6}$$

For the region containing π_0, let the minimizing alternative be a*. The associated α-vector is α^*, where

$$\alpha^* = \gamma^{a^*} + \beta \sum_{\theta} P^{a^*} R_{\theta} \alpha^{n-1}_{d(\pi_0,\theta,a^*)}$$

Determining the Region Containing π_0

The next step of the algorithm is to determine the region in Π for which α^* is the appropriate α-vector. As a preliminary, consider Figure 4, an illustration of a two-state process. Figure 4 shows the minimum cost function at time $n-1$. Assume the set of α-vectors at time $n-1$ consists of only two elements. The two linear segments correspond to $\pi\alpha^{n-1}_1$ and $\pi\alpha^{n-1}_2$.

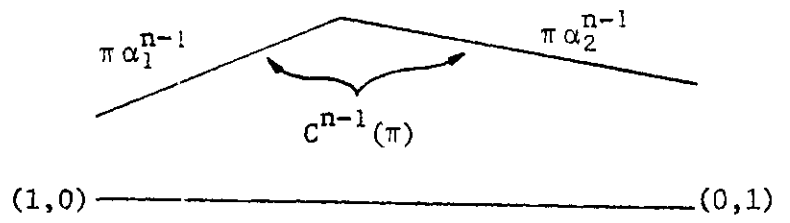
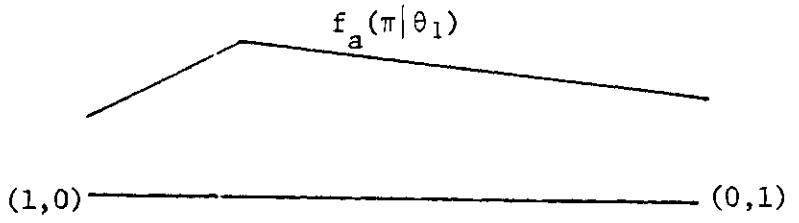


Figure 4. Cost Function for a Two-state Process at Time n - 1

For a specific alternative $a \in A$, define

$$f_a(\pi|\theta_i) = \min_j \beta \pi P^a R_{\theta_i} \alpha_j^{n-1}$$

where $f_a(\pi|\theta_i)$ is one of the terms under the summation on the right hand side of equation (4). Figures 5-a and 5-b show $f_a(\pi|\theta_1)$ and $f_a(\pi|\theta_2)$, respectively. Note that since there are only two linear segments in $C^{n-1}(\pi)$, there can only be two linear segments in $f_a(\pi|\theta_i)$. Figure 5-c illustrates $\sum_\theta f_a(\pi|\theta)$ which represents the total future cost for the remaining n - 1 periods.



5-a

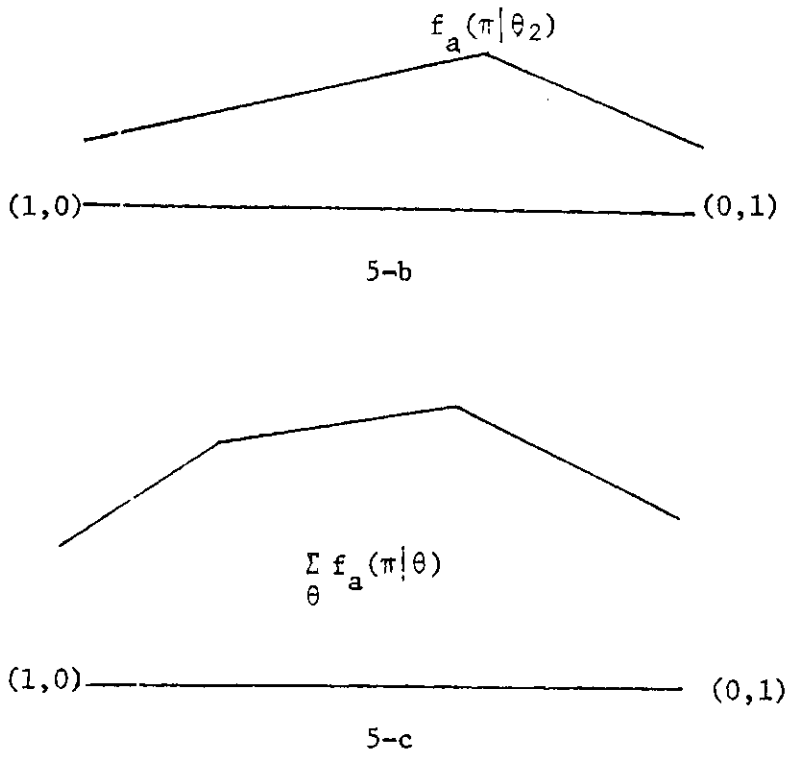


Figure 5. Future Cost for a Two-state Process at Time n

Define $C_a^n(\pi)$ as the cost function for alternative $a \in A$.

$$C_a^n(\pi) = \pi\gamma^a + \beta \sum_\theta \pi P^a R_\theta \alpha^{n-1}_{d(\pi,\theta,a)}$$

Figure 6 shows $C_a^n(\pi)$, the sum of the two functions $\pi\gamma^a$ and $\sum_\theta f_a(\pi|\theta)$. Note that $C_a^n(\pi)$ has the same piecewise linear, concave characteristics as $C^n(\pi)$.

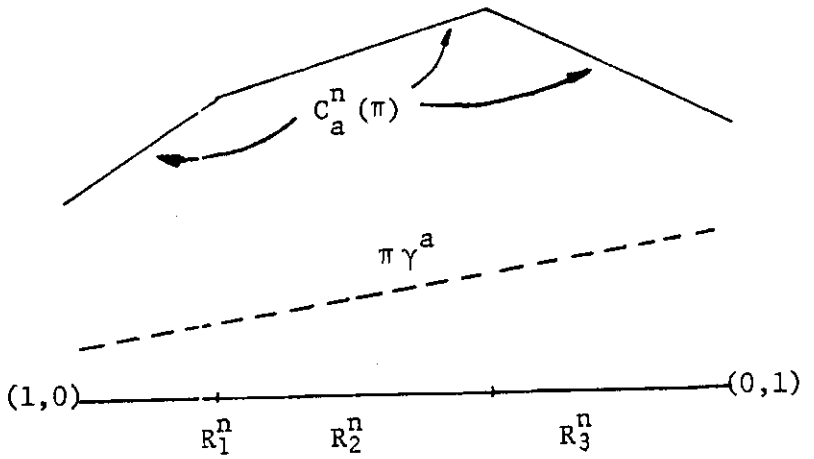


Figure 6. Cost Function for a Two-state Process at Time n for Alternative $a \in A$

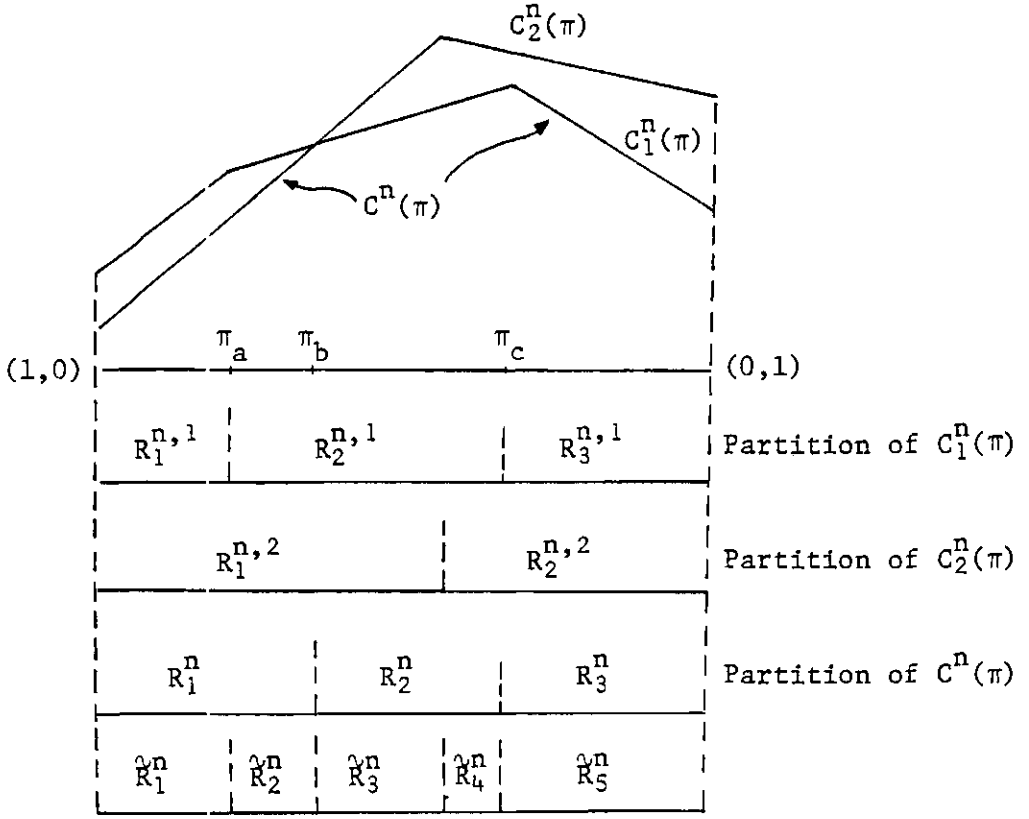


Figure 7. Cost Function for a Two-state Process at Time n

Consider Figure 7. This is an example of a two-state problem which has two available decision alternatives. The cost function $C_1^n(\pi)$ partitions the probability state space into three regions, $C_2^n(\pi)$ partitions it into two regions. The minimum of $C_1^n(\pi)$ and $C_2^n(\pi)$ is $C^n(\pi)$ and partitions the state space into three regions. Let the union of the boundaries of the partitions induced by the three cost functions form a new partition, $\tilde{R}^n = \{\tilde{R}_1^n, \tilde{R}_2^n, \ldots, \tilde{R}_5^n\}$. Each of the regions in $\tilde{R}^n$ differs from an adjacent region in one of the following ways:

1. The adjacent region has the same minimizing alternative, the same α-vector, but a different set of mappings, for example, adjacent regions $\tilde{R}_1^n$ and $\tilde{R}_2^n$.

2. The adjacent region has the same set of mappings but a different minimizing alternative and therefore a different α-vector, for example, adjacent regions $\tilde{R}_2^n$ and $\tilde{R}_3^n$.

3. The adjacent region has the same minimizing alternative, a different α-vector and a different set of mappings, for example, $\tilde{R}_4^n$ and $\tilde{R}_5^n$.

At this stage of the algorithm the region in $\tilde{R}^n$ containing π_0 is defined by finding all adjacent regions. To find these regions, first consider the region in Π which contains the initial point π_0, denoted $\tilde{R}_1^n$. Consider moving away from π_0 until some π is reached which is on the boundary of an adjacent region. There are three ways for this to occur.

In the first case, some π is reached such that the adjacent region has characteristic (1) as described above. For some alternative $a \neq a^*$, one item in the given set of mappings $\{(\theta,a) \rightarrow \alpha_j^{n-1}\}$ changes for some θ; that is, $d(\pi_0,\theta,a) \neq d(\pi,\theta,a)$. Point π_a in Figure 7 illustrates this.

Thus points in the region containing π_0 satisfy

$$\pi P^a R_\theta \alpha^{n-1}_{d(\pi_0,\theta,a)} \leq \pi P^a R_\theta \alpha^{n-1}_j \quad ;$$

$$j=1,2,\ldots,S^{(n-1)}$$
$$j\neq d(\pi_0,\theta,a)$$
$$\text{for all } \theta$$

or

$$\pi P^a R_\theta (\alpha^{n-1}_{d(\pi_0,\theta,a)} - \alpha^{n-1}_j) \leq 0$$

$$j=1,2,\ldots,S^{(n-1)}$$
$$j\neq d(\pi_0,\theta,a)$$
$$\text{for all } \theta$$

In the second case, some π is reached such that the adjacent region has characteristic (2) as described above. For the given set of mappings, the minimizing alternative in equation (6) changes; that is,

$$\pi \gamma^{a^*} + \beta \pi \sum_\Theta P^{a^*} R_\theta \alpha^{n-1}_{d(\pi_0,\theta,a^*)} \geq \pi \gamma^a + \beta \pi \sum_\theta P^a R_\theta \alpha^{n-1}_{d(\pi_0,\theta,a)}$$

Point π_b illustrates this occurrence. Points in the region containing π_0 satisfy

$$\pi \alpha^* - \pi \alpha_a \leq 0 \qquad \text{for all } a \neq a^*$$

where

$$\alpha_a = \gamma^a + \beta \sum_\theta P^a R_\theta \alpha^{n-1}_{d(\pi_0,\theta,a)}$$

In the third case, some π is reached such that the adjacent region

has characteristic (3) as described previously. For alternative a^*, one item in the given set of mappings $\{(\theta,a^*) \rightarrow \alpha_j^{n-1}\}$ changes for some θ; that is, $d(\pi_0,\theta,a^*) \neq d(\pi_c,\theta,a^*)$. Point π_c illustrates this. Points in the region containing π_c satisfy

$$\pi P^{a^*} R_\theta \alpha^{n-1}_{d(\pi_0,\theta,a^*)} \leq \pi P^{a^*} R_\theta \alpha_j^{n-1} \quad ;$$

$$j=1,\ldots,S^{(n-1)}$$
$$j \neq d(\pi_0,\theta,a^*)$$
$$\text{for all } \theta$$

or

$$\pi P^{a^*} R_\theta (\alpha^{n-1}_{d(\pi_0,\theta,a^*)} - \alpha_j^{n-1}) \leq 0$$

$$j=1,\ldots,S^{(n-1)}$$
$$j \neq d(\pi_0,\theta,a^*)$$
$$\text{for all } \theta$$

Setting up the Linear Program

The region containing π_0 is a convex region formed by the set of constraints

$$\pi(\alpha^* - \alpha_a) \leq 0 \qquad \text{for all } a \neq a^* \qquad (7)$$

$$\pi P^a R_\theta (\alpha^{n-1}_{d(\pi_0,\theta,a)} - \alpha_j^{n-1}) \leq 0 \qquad (8)$$

$$\text{for all } a \neq a^*$$
$$\text{for all } \theta$$
$$\text{for all } j \neq d(\pi_0,\theta,a)$$

$$\pi P^{a*} R_\theta \left(\alpha^{n-1}_{d(\pi_0,\theta,a*)} - \alpha^{n-1}_j\right) \leq 0 \qquad (9)$$

for all θ

for all $j \neq d(\pi_0,\theta,a*)$

along with

$$\sum_i \pi_i = 1$$

$$\pi_i \geq 0$$

Some of these constraints are unnecessary in that they are not part of the convex hull formed by the set of constraints. In order to find the convex hull of these constraints, let each of the constraints, f_i, in (7), (8), and (9) be, in turn, the objective function to the linear programming problem:

$$\text{max:} \quad f_i$$

$$\text{s.t.:} \quad \pi(\alpha* - \alpha_a) \leq 0 \quad \text{for all } a \neq a*$$

$$\pi P^{a} R_\theta \left(\alpha^{n-1}_{d(\pi_0,\theta,a)} - \alpha^{n-1}_j\right) \leq 0$$

for all $a \neq a*$

for all θ

for all $j \neq d(\pi_0,\theta,a)$

$$\pi P^{a*} R_\theta (\alpha^{n-1}_{d(\pi_0,\theta,a*)} - \alpha^{n-1}_j) \leq 0$$

for all θ
for all $j \neq d(\pi_0,\theta,a*)$

$$\sum_i \pi_i = 1$$

$$\pi_i \geq 0$$

If, at optimality, the objective function equals zero, then the corresponding constraint is part of the convex hull forming the convex region which contains π_0. For each constraint determined to be part of the convex hull, an adjacent region has been found. If it is a constraint from (7), the new region has the same set of mappings, but the α-vector changes to α_a and the minimizing alternative is a. If the constraint is from (8), the new region has the same α-vector, the same minimizing alternative, but one item in the set of mappings changes from $d(\pi_0,\theta,a)$ to j. For a constraint from (9), the new region has the same minimizing alternative, the new α-vector is $P^{a*}R_\theta\alpha^{n-1}_j$, and one item in the set of mappings changes from $d(\pi_0,\theta,a*)$ to j. Each time a new region is found, it is only necessary to record its minimizing alternative, its α-vector and its set of mappings.

The algorithm thus continues in the same fashion by considering any new region and its descriptive characteristics. The set of constraints associated with the region are formed and additional regions located. The entire set of α-vectors will be found by this method because the state space Π will be searched completely.

Recall that the objective of the algorithm is to find the set of α-vectors corresponding to the regions in the partition formed by $C^n(\pi)$. This is done simply by listing all the different α-vectors found in the above procedure, ignoring any differences in the mappings.

In summary, the major steps of the algorithm are as follows:

1. Choose an initial point π_0.

2. Determine $C^n(\pi_0)$. This establishes α^*, a^* and the set of mappings.

3. Place α^*, a^*, and the set of mappings of the first region on the list of regions to be searched.

4. Choose a region to be searched. Stop if the list is empty.

5. Set up the linear programming problem corresponding to that region and find all adjacent regions.

6. Add any new regions and their characteristics to the list and return to step 4.

CHAPTER IV

COST VARIANCE INVESTIGATION DECISIONS: THE INFINITE HORIZON CASE

The model presented in Chapter III for the investigation of cost variances is now extended to consider an infinite planning horizon. Situations in which the operating time until termination is unknown and where it is assumed to be large are in practice modeled as having an infinite number of time periods to operate. Kaplan [13] examines this extension and the solution by allowing the number of remaining time periods to increase until a steady state solution is found. However, this is a very inefficient approach for problems having more than two states and practically unusable where there are five or more states.

The assumptions for the infinite-horizon cost variance investigation problem are identical to those of the finite-horizon case, except for the operating time. As with the finite case, the true state of the process is unknown each time period except immediately following an investigation. The process is modeled as an infinite-horizon discounted partially observable Markov process where the state space is Π, representing knowledge of the true state of the underlying process.

Recall that $C^n(\pi)$ is the minimum cost for a process which has n remaining operating periods and terminal cost $C^0(\pi)$. Define $\delta^n(\pi)$ as the optimal policy with n remaining periods. Therefore

$$C^n(\pi) = \pi\gamma^{\delta^n(\pi)} + \beta \sum_{\theta} Pr\{\theta|\pi,\delta^n(\pi)\}C^{n-1}[T(\pi|\theta,\delta^n(\pi)]$$

where $0 < \beta \leq 1$.

Define $C^*(\pi)$ as the minimum expected cost of a process which has an infinite number of periods to operate. It can be shown that $C^*(\pi)$ satisfies the following recursion:

$$C^*(\pi) = \min_{a \in A} \{\pi \gamma^a + \beta \sum_{\theta} P\{\theta | \pi, a\} C^*[T(\pi | \theta, a)]\} \tag{10}$$

Let the sequence of policy rules for the infinite horizon policy be represented by $\{\delta^0, \delta^1, \ldots\}$. If the policy rule is constant for each time period, the policy is stationary and is denoted by $\delta^\infty = \{\delta, \delta, \ldots\}$.

Sondik's [26] algorithm for the infinite horizon problem with discounting can be used to solve cost-variance investigation problems over an infinite horizon. As with the finite case, the algorithm efficiently handles problems having a large number of states. The remainder of this chapter is a discussion of Sondik's algorithm for the infinite-horizon discounted partially observable Markov process, applied to the N_s-state cost-variance investigation problem.

Preliminaries

Certain well-known properties of the optimal control policy for the infinite-horizon problem with discounting are now presented. Proofs of these following results can be found in Sondik [26].

1. The minimum discounted expected cost, $C^*(\pi)$, of operating a process over an infinite number of time periods does exist.

2. The value of $C^*(\pi)$ is the limit of $C^n(\pi)$ as n approaches infinity; $C^*(\pi) = \lim_{n \to \infty} C^n(\pi)$.

3. $C^*(\pi)$ is independent of any terminal cost, $C^0(\pi)$.

4. The optimal control policy is a unique stationary policy, denoted $(\delta^*)^\infty$.

5. The total discounted expected cost of any stationary policy, δ^∞, is a well-defined function and is denoted by $C(\pi|\delta)$.

Since the optimal policy is known to be stationary, a reasonable choice for a policy rule would be $(\delta^n)^\infty$ for some large n, where δ^n is the policy that would be chosen given n periods to operate. The problem is to determine n large enough so that $C(\pi|\delta)$ is close to $C^*(\pi)$. Sondik [26] has shown that

$$\sup_{\pi} \left| C(\pi|\delta^n) - C^*(\pi) \right| \leq \frac{2\beta^n}{1-\beta} Q$$

where

$$Q = \max_{a,i} \left| \gamma_i^a \right|$$

Thus, for a given $\varepsilon > 0$, where ε is chosen so that

$$\sup_{\pi} \left| C(\pi|\delta^n) - C^*(\pi) \right| \leq \varepsilon ,$$

then

$$\frac{2\beta^n}{1-\beta} Q \leq \varepsilon$$

or

$$n \geq \frac{\ell n\left[\frac{\varepsilon(1-\beta)}{2Q}\right]}{\ell n \beta}$$

For values of β close to 1, the value of n becomes increasingly large. For example, if $Q = 3$, $\beta = 0.9$, and $\varepsilon = 10^{-3}$, then $n \geq 105$. For $\beta = 0.98$, $n \geq 625$.

As this illustrates, repeated application of Sondik's One-pass Algorithm is not a very practical method for the infinite-horizon problem.

The algorithm for the infinite horizon problem uses the One-pass Algorithm but in a different fashion from straightforward repeated application. The One-pass Algorithm is used in a variation of Howard's policy iteration algorithm for the completely observable Markov process. First the total discounted expected cost for a given stationary policy is found (value determination), then this cost is used in the One-pass Algorithm to seek a stationary policy with a lower cost (policy improvement). This is then repeated until a policy is found which is ε-optimal: that is, for some constant $\varepsilon > 0$, $\sup_{\pi} \left| C(\pi|\delta) - C^*(\pi) \right| < \varepsilon$.

Value Determination

In the value determination part of the algorithm, the total expected cost, $C(\pi|\delta)$, is found for a given stationary policy, δ^∞. Because the policy decision rule is stationary, the cost function is written

$$C(\pi|\delta) = \pi\gamma^{\delta(\pi)} + \beta \sum_{\theta} P\{\theta|\pi,\delta\} C[T(\pi|\theta,\delta)|\delta] \qquad (11)$$

Although $C^n(\pi|\delta)$ is piecewise linear, $C(\pi|\delta)$ is not necessarily so. This means that Sondik's algorithm as developed in Chapter III is not directly applicable.

Nonetheless, there is a class of stationary policies, called

finitely transient policies, for which the minimal expected discounted cost is in fact piecewise linear.

A policy is finitely transient if and only if there exists some positive interger, n_δ, such that, after n_δ transitions, there is no state $\pi \in \Pi$ which will lie in the set of states for which two or more distinct policies are optimal.

To amplify, define D_δ as the smallest closed set in Π for which the policy is discontinuous. For example, if for a two-state process, the policy rule is

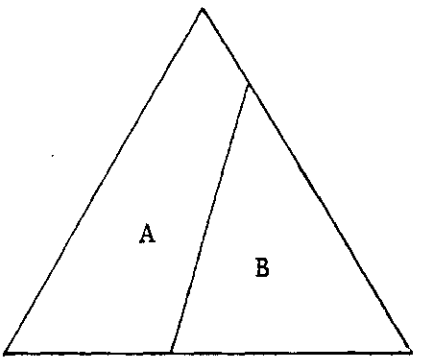
$$\pi \leq 0.4 \qquad \delta = a_1$$

$$\pi \geq 0.4 \qquad \delta = a_2$$

then $D_\delta = \{(.4,.6)\}$. For the three-state process illustrated below, the policy rule is

$$\pi \in A \qquad \delta = a_1$$

$$\pi \in B \qquad \delta = a_2$$



The set D_δ consists of the set of points which lie on the boundary separating A and B.

Consider a sequence of transitions for a value $\pi \in \Pi$. The definition of the operator T for the policy rule δ^∞ is as follows:

$$\pi' = T(\pi|\theta,\delta)$$

$$= \frac{\pi P^\delta R_\theta}{P\pi\{\theta|\pi,\delta\}}$$

For two transitions, the iterated use of the operator T is defined as

$$\pi^{(2)} = T(\pi|\theta_1,\theta_2,\delta) = T[T(\pi|\theta_1,\delta)|\theta_2,\delta]$$

For n_δ transitions

$$\pi^{(n_\delta)} = T\{T\ldots[T(\pi|\theta_1,\delta)|\theta_2,\delta]\ldots|\theta_{n_\delta},\delta\}$$

where $\pi^{(n_\delta)}$ is the value in Π which is the result of n_δ transitions, having received a sequence of cost observations, $\{\theta_1,\theta_2,\ldots\theta_{n_\delta}\}$. If δ^∞ is finitely transient with index n_δ, then for every $\pi \in \Pi$,

$$\pi^{(n_\delta)} \notin D_\delta$$

Finitely transient policies are useful for two reasons:

1. The cost of a finitely transient policy can be readily calculated using nearly the same method of computing the cost as for a completely observable Markov process.

2. Any stationary policy has properties closely resembling a finitely transient policy; thus its cost can be approximated by using the finitely transient policy closely resembling it.

For every policy which is finitely transient, there exists a finite partition $V = \{V_1, V_2, \ldots, V_M\}$ of the probability state space induced by the policy δ^∞. This partition is the key which allows Howard's policy iteration algorithm to be applied to the infinite horizon investigative problem.

To establish the partition, define the set D^1 as follows:

$$D^1 = [\pi : T(\pi | \theta, \delta) \in D_\delta]$$

That is, D^1 is the set of values $\pi \in \Pi$ such that their transition is in D_δ. In like manner, define D^2 as

$$D^2 = [\pi : T(\pi | \theta, \delta) \in D^1]$$

Clearly, for $\pi \in D^2$ and sequence of cost observations, $\{\theta_1, \theta_2\}$,

$$T(\pi | \theta_1, \theta_2, \delta) \in D_\delta$$

In general,

$$D^n = [\pi : T(\pi | \theta, \delta) \in D^{n-1}]$$

After n_δ transitions

$$D^{n\delta} = [\pi : T(\pi | \theta, \delta) \in D^{n\delta - 1}]$$

is empty.

A partition is formed for the policy δ^∞ by finding the sets D^δ, $D^1, D^2, \ldots, D^{n_\delta - 1}$, which can be identified as boundaries of regions in the probability state space. A detailed discussion of the formation of the partition is found in Appendix B.

Given that δ^∞ is finitely transient, the finite partition $V = \{V_j\}$ satisfies the following properties.

1. If π_1 and $\pi_2 \in V_j$, then $\delta(\pi_1) = \delta(\pi_2) = \delta_j$; that is, any two states in the same region have the same action.

2. A mapping ν exists for the policy δ such that, for a given θ, if $\pi \in V_j$, then $T(\pi|\theta,\delta) \in V_{\nu(j,\theta)}$; that is, all states in region j, for a given θ, map into the same region $V_{\nu(j,\theta)}$ after one transition.

The result of this property is that for a given $\pi \in \Pi$, $C(\pi|\delta)$ can be written.

$$C(\pi|\delta) = \pi\,\alpha(\pi|\delta) \tag{12}$$

where the vector $\alpha(\pi|\delta)$ is the unique bounded solution to the equation

$$\alpha(\pi|\delta) = \gamma^{\delta(\pi)} + \beta \sum_\theta P^{\delta(\pi)} R_\theta\, \alpha[T(\pi|\theta,\delta)|\delta] \tag{13}$$

For $\pi \in V_j$, define $\alpha(\pi|\delta) = \alpha_j$. Then for $\delta(\pi) = \delta_j$, equation (13) is

$$\alpha_j = \gamma^{\delta_j} + \beta \sum_\theta P^{\delta_j} R_\theta\, \alpha_{V_{\nu(j,\theta)}} \tag{14}$$

Thus

$$C(\pi|\delta) = \pi\,\alpha_j \tag{15}$$

for every $\pi \in V_j$. Therefore it follows that $C(\pi|\delta)$ is piecewise linear. For a more complete discussion and proofs of the properties of finitely transient policies, see Sondik [26].

The partition $V = \{V_1,\ldots V_M\}$ and the piecewise linearity of the cost function are illustrated in Figure 8. This is similar to the illustration of the finite horizon case shown in Figure 3, with the exception that the cost function for the infinite horizon problem is not necessarily continuous.

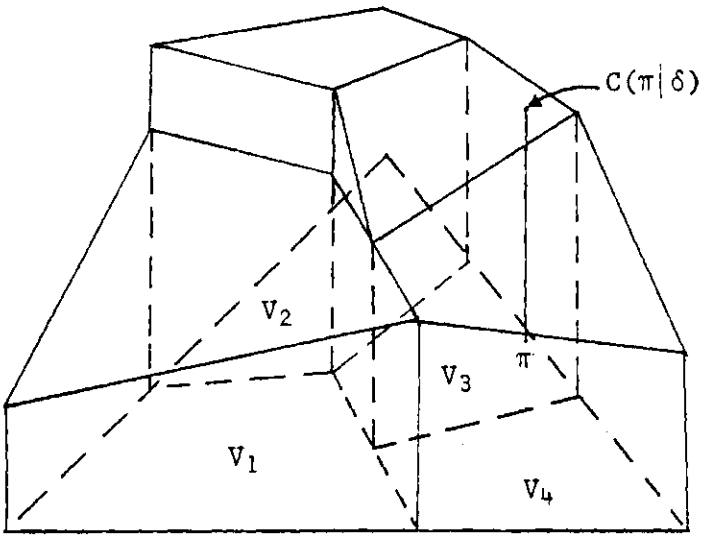


Figure 8. Cost Function over an Infinite Horizon

Equations (14) and (15) show that the cost function has a simple form, which leads to the development of a procedure for the calculation of $C(\pi|\delta)$. Each region $V_j \in V$ has an α-vector associated with it from the set $\{\alpha_1,\alpha_2,\ldots\alpha_M\}$, where the set of α-vectors is calculated from (14). As an example, in Figure 8, if $\pi \in V_3$, if $T(\pi|\theta_1,\delta) \in V_4$ and $T(\pi|\theta_2,\delta) \in V_2$,

then from (14),

$$\alpha_3 = \gamma^{\delta_3} + \beta(P^{\delta_3} R_{\theta_1} \alpha_4 + P^{\delta_3} R_{\theta_2} \alpha_2)$$

and one of the recursive equations necessary for the complete solution of the set of α-vectors has been formed.

In setting up the entire set of recursive equations to solve for the set of α-vectors, it is only necessary to know (1) the number of regions in the partition of the probability state space, (2) the decision alternative δ_j associated with each region, and (3) the mapping $\nu(j,\theta)$ of each pair (δ_j,θ) onto the set of regions in the partition, for one point in each region V_j.

In matrix form the set of recursive equations can be written as

$$\bar{\alpha} = \bar{\gamma} + \beta \bar{P} \bar{\alpha} \tag{16}$$

where $\bar{\alpha}$, $\bar{\gamma}$ and $\bar{P}$ have the following definitions. Note that (16) has the same form as the matrix equation for expected cost in a completely observable Markov process.

Given M regions in the partition, let $\bar{\alpha}$ be an $M \cdot N_s$ vector such that

$$\bar{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_M \end{bmatrix}$$

Let

$$\overline{\gamma} = \begin{bmatrix} \gamma^{\delta_1} \\ \gamma^{\delta_2} \\ \vdots \\ \gamma^{\delta_M} \end{bmatrix}$$

Let

$$\overline{P} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1,M} \\ A_{21} & A_{22} & \cdots & A_{2,M} \\ \vdots & \vdots & & \vdots \\ A_{M,1} & A_{M,2} & \cdots & A_{M,M} \end{bmatrix}$$

where A_{jk} either equals $P^{\delta_j}R_\theta$ if $\nu(j,\theta) = k$, or is a matrix of zeros if $\nu(j,\theta) \neq k$.

As an example, assume that the given policy is finitely transient, having six regions in the partition. There are two θ values and two decision alternatives. Let the mapping, ν, be as follows:

j	δ_j	$\nu(j,\theta_1)$	$\nu(j,\theta_2)$
1	2	3	4
2	2	3	5
3	2	6	5
4	2	1	1
5	1	3	2
6	1	1	2

Thus $\overline{P}$ is given by

$$\overline{P} = \begin{bmatrix} 0 & 0 & P^2R_1 & P^2R_2 & 0 & 0 \\ 0 & 0 & P^2R_1 & 0 & P^2R_2 & 0 \\ 0 & 0 & 0 & 0 & P^2R_2 & P^2R_1 \\ P^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & P^1R_2 & P^1R_1 & 0 & 0 & 0 \\ P^1R_1 & P^1R_2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

From equation (16), $\overline{\alpha}$ is calculated as

$$\overline{\alpha} = (I - \beta \overline{P})^{-1} \overline{\gamma}$$

where $\overline{\alpha}$ is the vector of α-vectors. This establishes the solution for the total discounted expected cost for finitely transient policy δ^∞.

Policy Improvement

Given the total expected discounted cost for a stationary policy, δ^∞, the next part of the algorithm is to seek a new policy $(\delta^1)^\infty$ which has a lower expected cost than the previous one. If $C(\pi|\delta)$ is continuous and concave, this could be accomplished by finding

$$C(\pi|\delta^1) = \min_a \{\pi\gamma^a + \sum_\theta P\{\theta|\pi,a\} C[T(\pi|\theta,a)|\delta]$$

Even though $C(\pi|\delta)$ is piecewise linear, it may not necessarily be continuous. However, there always exists a concave hull, $\overline{C}(\pi|\delta)$, of $C(\pi|\delta)$ which is a continuous function. An example of the concave hull of a non-

continuous function is shown in Figure 9.

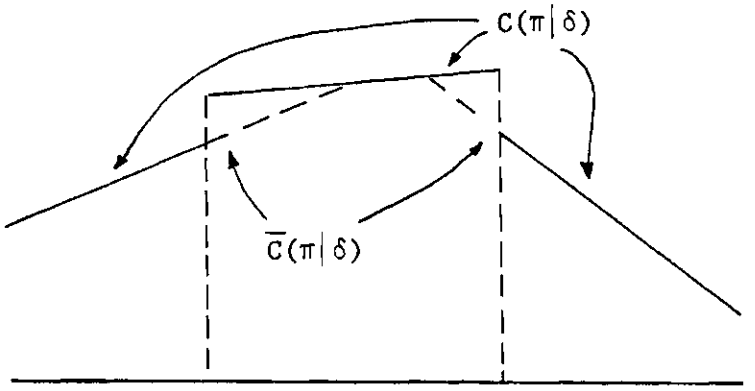


Figure 9. Concave Hull of a Non-continuous Function

The concave hull $\overline{C}(\pi|\delta)$ may be calculated from the set of α-vectors for $C(\pi|\delta)$, without explicit reference to π.

$$\overline{C}(\pi|\delta) = \min_{k}\{\pi\alpha^{k}\} \qquad k=1,\ldots,M$$

Thus a new control policy $(\delta^1)^\infty$ can be calculated from $\overline{C}(\pi|\delta)$ using the One-pass Algorithm. The policy $(\delta^1)^\infty$ is represented by a new set of α-vectors.

To determine if optimality for the process has been reached closely enough for the iterative procedure to be terminated, an error bound must be established. Optimality is not guaranteed after a finite number of iterations for this partially observable Markov process, since the state space is infinite. Sondik [26] has shown that the upper bound on the difference between the cost of the concave hull of the stationary policy and the true optimal policy is the following:

$$\sup_{\pi}|\overline{C}(\pi|\delta)-C^*(\pi)| \leq \frac{1}{1-\beta}\sup_{\pi}|\overline{C}(\pi|\delta)-C(\pi|\delta^1)| \qquad (17)$$

Therefore every time a set of α-vectors for a new policy has been found, the stopping rule given by (17) is used. When $\frac{1}{1-\beta}\sup_{\pi}|\overline{C}(\pi|\delta)-C(\pi|\delta^1)|$ is sufficiently small, the iterative procedure ends.

Sondik's Approximation Method

The entire procedure presented above can be used even when policies are not finitely transient. An approximate solution can be found by replacing the given stationary policy with the finitely transient policy which closely resembles it. Assume that a partition V^k has been calculated for policy δ^∞ from a sequence of sets D_δ, $D^1, D^2, \ldots, D^k$. If $D^k=\Phi$, the empty set, then policy δ^∞ is finitely transient. If $D^k\neq\Phi$, then it is still not known whether δ^∞ is finitely transient or not. Assume $D^k\neq\Phi$. The partition V^k can be used for approximating the cost function of the policy δ^∞. Given the partition, an approximation mapping, $\hat{\nu}$, can be determined. For each $V_j^k \in V^k$, choose a state $\pi \in V_j^k$ and define the mapping as follows. For $\pi \in V_j^k$, if $T(\pi|\theta,\delta) \in V_\ell^k$, then $\hat{\nu}(j,\theta) = \ell$. However, since V^k is an approximate partition for δ^∞, there will exist, for some $V_j^k \in V^k$, a state π_1 such that

$$T(\pi_1|\theta,\delta) \notin V_{\hat{\nu}(j,\theta)}$$

An approximation of $C(\pi|\delta)$, denoted $\hat{C}(\pi|\delta)$ is calculated from V^k and $\hat{\nu}$. Thus $\hat{C}(\pi|\delta) = \pi\hat{\alpha}_j$ for $\pi \in V_j^k$, where $\hat{\alpha}_j$ is found from the set of recursive equations

$$\{\hat{\alpha} = \gamma^{\delta j} + \beta \sum_{\theta} P^{\delta j} R_{\theta} \hat{\alpha}_{\hat{\nu}(j,\theta)}\}$$

Since $\hat{C}(\pi|\delta)$ is an approximation to $C(\pi|\delta)$, a bound on the difference must be established. This difference has been shown [26] to satisfy

$$\sup_{\pi}|C(\pi|\delta)-\hat{C}(\pi|\delta)| \leq \frac{\beta}{1-\beta}\{\sup_{\pi}\sum_{\theta} P_r\{\theta|\pi,\delta\}T(\pi|\theta,\delta)[\hat{\alpha}_{\mu[T(\pi|\theta,\delta]} - \hat{\alpha}_{\hat{\nu}(j,\theta)}]\}$$

where $\mu[T(\pi|\theta,\delta)]$ is defined as the index of the set V^k which contains $T(\pi|\theta,\delta)$. If the policy δ^{∞} is finitely transient and V^k is the equivalent partition, then

$$\mu[T(\pi|\theta,\delta)] = \hat{\nu}(j,\theta)$$

Before the partition V^k can be determined, the value of k, the index of approximation, must be chosen. It is possible to establish values of k so that the error on the difference between the approximate and the exact cost is guaranteed to be within given limits. However, Sondik suggests choosing k equal to 3 or 4, claiming that it leads to a fairly good approximation. After the set of vectors $\{\hat{\alpha}_1,\hat{\alpha}_2,\ldots,\hat{\alpha}_{M_k}\}$ has been calculated, the error can be determined and if it is too great then k can be increased by 1 and the procedure repeated.

The basic steps of the algorithm are as follows:

1. Choose an initial control policy δ^{∞}; for example, the policy which gives the minimum immediate expected cost.

2. Establish k; for example, let k=4. Find the partition V^k and

the mapping $\hat{\nu}$.

3. Calculate the cost function $\hat{C}(\pi|\delta) = \pi\hat{\alpha}_j$ for $\pi \in V_j^k$, where $\hat{\alpha}_j = \gamma^{\delta j} + \beta\Sigma_\theta P^{\delta j}_\theta R_\theta \hat{\alpha}_{\hat{\nu}(j,\theta)}$. For $\varepsilon_1>0$, if $\sup_\pi |\hat{C}(\pi|\delta)-C(\pi|\delta)| < \varepsilon_1$, go to step 4. If not, increase k by 1 and return to step 2.

4. Find a new control policy δ^1 from

$$C(\pi|\delta^1) = \min_a \{\pi\gamma^a + \beta\Sigma_\theta P_\pi\{\theta|\pi,a\}\bar{\hat{C}}[T(\pi|\theta,\delta)|\delta]\}$$

where $\bar{\hat{C}}$ is the concave hull of $\hat{C}$. For $\varepsilon>0$, if $\sup_\pi |\bar{\hat{C}}(\pi|\delta)-C(\pi|\delta^1)| < \varepsilon$, stop. Policy (δ^1) is ε-optimal. Otherwise let $\delta^\infty=(\delta^1)^\infty$ and return to step 2.

A Modified Approximation Method

The implementation of Sondik's algorithm can be simplified even beyond his claim of choosing k equal to 3 or 4, by eliminating the step of finding and using the partition, V^k. Instead use the simple partition $V^0 = \{V_1^0, V_2^0, \ldots, V_{M_0}^0\}$ induced by the policy rule δ^∞. Unless the policy happens to be finitely transient with index $n_\delta = 1$, the partition V^0 will not be equivalent to the policy δ^∞. Therefore the mapping, $\hat{\nu}(j,\theta)$, of each pair (δ_j,θ) onto the set of regions in the partition V^0 will be approximate. An approximation of $C(\pi|\delta)$, $\hat{C}(\pi|\delta)$, is calculated from the recursive equations using V^0 and $\hat{\nu}$. No error checking procedure is performed at this point as with Sondik's method.

A new control policy δ^1 is now calculated in the policy improvement routine and optimality is determined by using the stopping rule (17). Stop if δ^1 is optimal; otherwise repeat the procedure using δ^1 unless cycling has occurred.

In using V^0, the phenomenon of cycling between two non-optimal

policies may be encountered. To break from the cycle, apply the One-pass Algorithm once to the policy δ^1 to yield a new policy. This, in effect, creates the partition $V^1 = \{V_1^1, V_2^1, \ldots, V_{M_1}^1\}$. Repeat the steps of the algorithm with δ^∞ replaced by the new stationary policy. If cycling continues after the first application (i.e., a second time), the One-pass Algorithm must be applied twice to break the cycle. In general, continue to increase the number of applications of the One-pass Algorithm to δ^1 by one until the cycle is broken.

Briefly, the argument for using this procedure over Sondik's approximation method is that even though the optimal policy might be obtained in a few iterations, a significant amount of computational work is required at each iteration of the algorithm; the search procedures are the most time consuming part of the computer routine. With the modified method, a greater number of iterations is usually required, but since the number of search procedures is reduced in eliminating the full treatment of the partition, the overall computer time is reduced. It must be recognized, of course, that this modification to Sondik's algorithm is a heuristic and has not been proven in general. Furthermore, the computational efficiencies realized in limited testing may not be achieved in all problem settings.

The basic steps of the modified algorithm are as follows:

1. Choose an initial policy δ^∞. Let $k = 0$.
2. Find the partition V^0 and the mapping $\hat{\nu}$.
3. Calculate the set of $\hat{\alpha}$-vectors using V^0 and $\hat{\nu}$.
4. Find a new control policy δ^1 from

$$C(\pi|\delta^1) = \min_a \{\pi\gamma^a + \beta\Sigma_\theta P\prime\{\theta|\pi,a\}\bar{\hat{C}}[T(\pi|\theta,\delta|\delta]$$

where $\bar{\hat{C}}$ is the concave hull of $\hat{C}$. For $\varepsilon>0$, if $\sup_{\pi}|C(\pi|\delta)-C(\pi|\delta^1)| < \varepsilon$, then stop. Policy δ^1 is ε-optimal. If cycling has occurred go to step 5. Otherwise let $\delta = \delta^1$ and return to step 2.

5. Let $k = k+1$. Find a new control policy $(\delta^1)^k$, by applying the One-pass Algorithm k times to δ^1. Let $\delta = (\delta^1)^k$ and return to step 2.

CHAPTER V

RESULTS

Finite Horizon Example

As an illustration of the finite horizon algorithm, consider the following cost variance investigation problem. The process can be in one of three states, (1) an in-control state, (2) an intermediate state, and (3) an out-of-control state. The end-of-period cost report is one of two values, 0 and 3. There are two decision alternatives available, (1) investigate and (2) do not investigate. The cost of investigation is constant for all states, $K = 1$. The transition probabilities, the cost observation probabilities and other parameters are given in Table 2.

Table 2. Parameters for a Three-state Cost Variance Investigation Problem

a_1: Investigate

$$P^1 = \begin{bmatrix} .8 & .15 & .05 \\ .8 & .15 & .05 \\ .8 & .15 & .05 \end{bmatrix}$$

a_2: Do not Investigate

$$P^2 = \begin{bmatrix} .8 & .15 & .05 \\ 0 & .8 & .2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} .8 & .2 \\ .6 & .4 \\ 0 & 1 \end{bmatrix}$$

$$C^\circ(\pi) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \qquad \beta = 0.9$$

The immediate expected cost vectors are $\gamma^a = [\gamma_i^a]$, where $\gamma_i^{a_1} = K + \sum_{j,\theta} P_{ij} r_{j\theta} \theta$ and $\gamma_i^{a_2} = \sum_{j,\theta} P_{ij} r_{j\theta} \theta$. Therefore

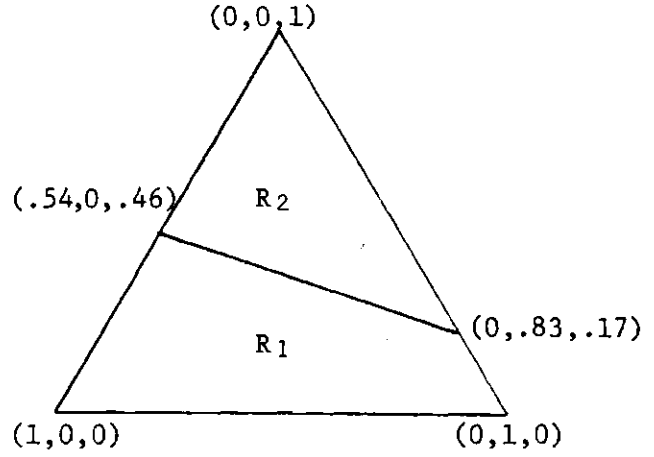
$$\gamma^1 = \begin{bmatrix} 1.81 \\ 1.81 \\ 1.81 \end{bmatrix} \quad \text{and} \quad \gamma^2 = \begin{bmatrix} 0.81 \\ 1.56 \\ 3.00 \end{bmatrix}$$

There are 10 periods remaining until termination.

Since the terminal cost for $n = 0$ is 0 for all states, the α-vectors for $n = 1$ are the immediate expected cost vectors. Therefore,

$$\alpha_1^1 = \begin{bmatrix} 0.81 \\ 1.56 \\ 3.00 \end{bmatrix} ; \delta_1^1 = 2 \quad \text{and} \quad \alpha_2^1 = \begin{bmatrix} 1.81 \\ 1.81 \\ 1.81 \end{bmatrix} ; \delta_2^1 = 1$$

Graphically, the solution to the problem for $n = 1$ is as follows:



$$R_1^1 : \alpha_1^1 = \begin{bmatrix} 0.81 \\ 1.56 \\ 3.00 \end{bmatrix} ; \delta_1^1 = 2$$

$$R_2^1 : \alpha_2^1 = \begin{bmatrix} 1.81 \\ 1.81 \\ 1.81 \end{bmatrix} ; \delta_2^1 = 1$$

To calculate the α-vectors for $n = 2$, first choose an initial $\pi_0 = (1,0,0)$ for region R_1^2. For each pair (θ,a) determine the mapping $d(\pi_0,\theta,a)$ onto the set of α-vectors, $\alpha^1 = [\alpha_1^1,\alpha_2^1]$ from $\min_k \pi_0 P^a R_\theta \alpha_k^{n-1}$. The mapping is

a \ θ	0	3
a_1	1	1
a_2	1	1

$C^2(\pi_0)$ is found from

$$C^2(\pi_0) = \min_a \{\pi_0 \gamma^a + \beta \sum_\theta P^a R_\theta \alpha^1_{d(\pi_0,\theta,a)}\}$$

$$= \min_a \{\pi_0 \alpha_{j=1}^{n=2,a=1}, \pi_0 \alpha_{j=1}^{n=2,a=2}\}$$

where

$$\alpha_{j=1}^{n=2,a=1} = \alpha_1^{2,1} = \pi_0 \gamma^1 + \beta \sum_\theta P^1 R_\theta \alpha^1_{d(\pi_0,\theta,1)}$$

$$d(\pi_0,\theta_1,1) = 1$$

$$d(\pi_0,\theta_2,1) = 1$$

and

$$\alpha_{j=1}^{n=2,a=2} = \alpha_1^{2,2} = \pi_0 \gamma^2 + \beta \sum_\theta P^2 R_\theta \alpha^1_{d(\pi_0,\theta,2)}$$

$$d(\pi_0,\theta_i,2) = 1$$

$$d(\pi_0,\theta_2,2) = 1$$

$$C^2(\pi_0) = \min\left\{\pi_0\begin{pmatrix}2.739\\2.739\\2.739\end{pmatrix}, \pi_0\begin{pmatrix}1.739\\3.223\\5.700\end{pmatrix}\right\}$$

Therefore,

$$C^2(\pi_0) = 1.739$$

and

$$\alpha_1^2 = \alpha_1^{2,2} = \begin{pmatrix}1.739\\3.223\\5.700\end{pmatrix} ; \delta_1^2 = 2$$

The regions adjacent to R_1^2 are found using the following set of constraints in a linear programming problem.

(a) $\pi(\alpha_1^2-\alpha_1^{2,1}) \leq 0$

(b) $\pi P^1 R_1(\alpha_1^1-\alpha_2^1) \leq 0$

(c) $\pi P^1 R_2(\alpha_1^1-\alpha_2^1) \leq 0$

(d) $\pi P^2 R_1(\alpha_1^1-\alpha_2^1) \leq 0$

(e) $\pi P^2 R_2(\alpha_1^1-\alpha_2^1) \leq 0$

$$\sum_i \pi_i = 1$$

$$\pi_i \geq 0$$

Using each constraint (a) - (e) in turn as the objective function

to the above set of constraints, it is found that only constraint (e) is tight. Therefore one new region R_2^2 is found and is described by

$$\alpha_2^2 = \alpha_1^{2,2} = \alpha_1^2 + \beta(P^2 R_2 \alpha_2^1 - P^2 R_2 \alpha_1^1) = \begin{bmatrix} 1.843 \\ 3.081 \\ 4.629 \end{bmatrix} ;$$

$$\delta_2^2 = 2$$

The mapping is

a \ θ	0	3
a_1	1	1
a_2	1	2

The non-optimal alternative is a = 1 and $\alpha_2^{2,1} = \begin{pmatrix} 2.739 \\ 2.739 \\ 2.739 \end{pmatrix}$.

Region R_2^2 is searched for its adjacent regions using the following set of constraints.

(a) $\pi(\alpha_2^2 - \alpha_2^{2,1}) \leq 0$

(b) $\pi P^1 R_1(\alpha_1^1 - \alpha_2^1) \leq 0$

(c) $\pi P^1 R_2(\alpha_1^1 - \alpha_2^1) \leq 0$

(d) $\pi P^2 R_1(\alpha_1^1 - \alpha_2^1) \leq 0$

(e) $\pi P^2 R_2(\alpha_2^1 - \alpha_1^1) \leq 0$

$$\sum_i \pi_i = 1$$

$$\pi_i \geq 0$$

Constraints (a) and (e) are found to be tight. Constraint (a) yields a new region R_3^2 for which

$$\alpha_3^{2,1} = \alpha_3^2 = \begin{pmatrix} 2.739 \\ 2.739 \\ 2.739 \end{pmatrix} ;$$

$$\delta_3^2 = 1$$

The mapping for R_3^2 is the same as for R_2^2. The non-optimal alternative is a = 2 and

$$\alpha_3^{2,2} = \begin{pmatrix} 1.843 \\ 3.081 \\ 4.629 \end{pmatrix}$$

Constraint (e) yields a region with

$$\alpha_4^2 = \alpha_2^2 + \beta(P^2R_2\alpha_2^1 - P^2R_2\alpha_1^1) = \begin{pmatrix} 1.739 \\ 3.223 \\ 5.700 \end{pmatrix} ;$$

$$\delta_4^2 = 2$$

and mapping

a \ θ	0	3
a_1	1	1
a_2	1	1

That is, however, the same set of characteristics describing region R_1^2 and is discarded. Therefore, the search of R_2^2 yields only one additional region.

The search of R_3^2 is accomplished using constraints

(a) $\pi(\alpha_3^2 - \alpha_3^{2,2}) \leq 0$

(b) $\pi P^1 R_1(\alpha_1^1 - \alpha_2^1) \leq 0$

(c) $\pi P^1 R_2(\alpha_1^1 - \alpha_2^1) \leq 0$

(d) $\pi P^2 R_1(\alpha_1^1 - \alpha_2^1) \leq 0$

(e) $\pi P^2 R_2(\alpha_2^1 - \alpha_1^1) \leq 0$

$$\sum_i \pi_i = 1$$

$$\pi_i \geq 0$$

Constraints (a) and (d) are tight. Constraint (a) returns to region R_2^2, thus yielding no additional information. Constraint (d) locates region R_4^2. Note that even though the alternative associated with constraint (d) is $a = 2$, the optimal alternative for the region being searched is $a = 1$. Therefore the α-vector describing region R_4^2 is the same as the region being searched.

$$\alpha_4^2 = \begin{pmatrix} 2.739 \\ 2.739 \\ 2.739 \end{pmatrix}; \quad \delta_4^2 = 1$$

The mapping is

a \ θ	0	3
a_1	1	1
a_2	2	2

The non-optimal alternative is a = 2 and

$$\alpha_4^{2,2} = \alpha_3^{2,2} + \beta P^{2}{}_{\cdot} R_1(\alpha_2^1 - \alpha_1^1) = \begin{pmatrix} 2.439 \\ 3.189 \\ 4.629 \end{pmatrix}$$

The search of region R_4^2 yields no new regions. Thus the entire set of α-vectors for n = 2 is

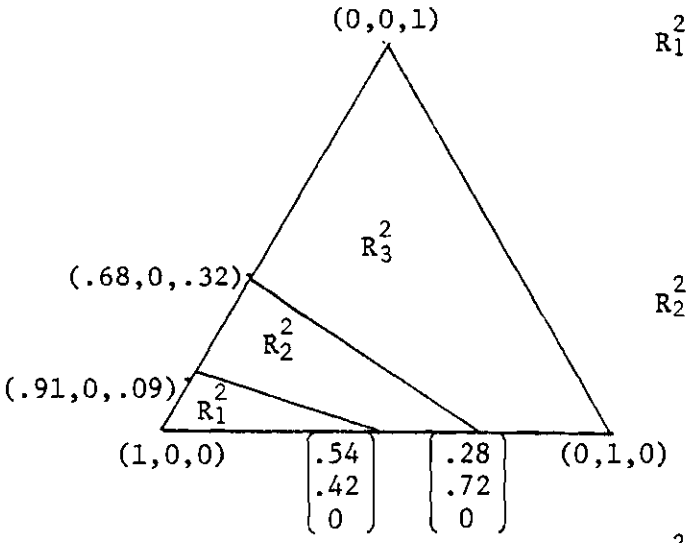
$$\alpha_1^2 = \begin{pmatrix} 1.739 \\ 3.223 \\ 5.700 \end{pmatrix}; \quad \delta_1^2 = 2$$

$$\alpha_2^2 = \begin{pmatrix} 1.843 \\ 3.081 \\ 4.629 \end{pmatrix}; \quad \delta_2^2 = 2$$

$$\alpha_3^2 = \begin{bmatrix} 2.739 \\ 2.739 \\ 2.739 \end{bmatrix} ; \delta_3^2 = 1$$

Note that there are only three distinct regions in the final solution as opposed to four regions following the search procedure. This is due to the fact that in the optimal solution for each time period, it is not necessary to distinguish between regions that differ only with respect to their mapping.

Graphically the solution for n = 2 is



$$R_1^2:\alpha_1^2 = \begin{bmatrix} 1.739 \\ 3.223 \\ 5.700 \end{bmatrix} ; \delta_1^2 = 2$$

$$R_2^2:\alpha_2^2 = \begin{bmatrix} 1.843 \\ 3.081 \\ 4.629 \end{bmatrix} ; \delta_2^2 = 2$$

$$R_3^2:\alpha_3^2 = \begin{bmatrix} 2.739 \\ 2.739 \\ 2.739 \end{bmatrix} ; \delta_3^2 = 1$$

Table 3 gives the complete solution to the example for n = 1 through n = 10.

Table 3. The Solution for the Finite Horizon Example Through Ten Time Periods

Time					
1	Decision	2	1		
	α-vectors	.810	1.810		
		1.560	1.810		
		3.000	1.810		
2	Decision	2	2	1	
	α-vectors	1.739	1.843	2.739	
		3.223	3.081	2.739	
		5.700	4.629	2.739	
3	Decision	2	2	1	2
	α-vectors	2.713	2.738	3.713	2.787
		4.673	4.234	3.713	4.173
		7.166	5.465	3.713	5.465
4	Decision	2	2	1	2
	α-vectors	3.599	3.632	4.599	3.655
		5.592	5.127	4.599	5.100
		7.918	6.341	4.599	6.341
5	Decision	2	2	1	2
	α-vectors	4.403	4.435	5.403	4.446
		6.393	5.927	5.403	5.916
		8.707	7.139	5.403	7.139
6	Decision	2	2	1	2
	α-vectors	5.125	5.158	6.125	5.163
		7.113	6.649	6.125	6.644
		9.425	7.862	6.125	7.862
7	Decision	2	2	1	2
	α-vectors	5.775	5.808	6.775	5.810
		7.763	7.299	6.775	7.297
		10.076	8.512	6.775	8.512
8	Decision	2	2	1	2
	α-vectors	6.360	6.393	7.360	6.394
		8.347	7.884	7.360	7.883
		10.661	9.097	7.360	9.097
9	Decision	2	2	1	2
	α-vectors	6.886	6.919	7.886	6.920
		8.874	8.410	7.886	8.410
		11.188	9.624	7.886	9.624
10	Decision	2	2	1	
	α-vectors	7.360	7.393	8.360	
		9.348	8.884	8.360	
		11.661	10.098	8.360	

Infinite Horizon Example

To illustrate the algorithm for the infinite horizon, consider the example problem presented in the preceding section. The parameter values are the same as for the finite horizon which are shown in Table 2.

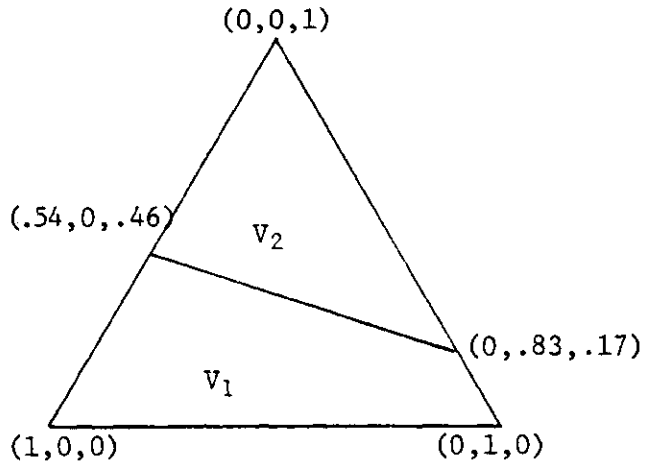
Let the initial stationary policy correspond to the minimum immediate expected cost. Therefore,

$$\alpha_1 = \begin{pmatrix} .81 \\ 1.56 \\ 3.00 \end{pmatrix} ; \delta_1 = 2$$

and

$$\alpha_2 = \begin{pmatrix} 1.81 \\ 1.81 \\ 1.81 \end{pmatrix} ; \delta_2 = 1$$

Graphically the partition induced on the probability state space is as follows:



$$V_1:\alpha_1 = \begin{pmatrix} .81 \\ 1.56 \\ 3.00 \end{pmatrix} ; \delta_1 = 2$$

$$V_2:\alpha_2 = \begin{pmatrix} 1.81 \\ 1.81 \\ 1.81 \end{pmatrix} ; \delta_2 = 1$$

Value Determination

To calculate the $\hat{\alpha}$-vectors, first choose a point in each region, find the transformed value for each θ and determine the region in which the transformed value is located. For the given policy, this is:

	π	$T(\pi\|\theta_1,\delta)$	$\hat{v}(j,\theta_1)$	$T(\pi\|\theta_2,\delta)$	$\hat{v}(j,\theta_2)$
V_1	(1,0,0)	(.877,.123,0)	1	(.593,.222,.185)	1
V_2	(.539,0,.461)	(.877,.123,0)	1	(.593,.222,.185)	1

To set up the recursive equations, form $\overline{\gamma}$ and $\overline{P}$.

$$\overline{\gamma} = \begin{bmatrix} .81 \\ 1.56 \\ 3.00 \\ 1.81 \\ 1.81 \\ 1.81 \end{bmatrix}$$

$$\overline{P} = \begin{bmatrix} P^2 & 0 \\ P^1 & 0 \end{bmatrix} = \begin{bmatrix} .8 & .15 & .05 & 0 & 0 & 0 \\ 0 & .8 & .2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ .8 & .15 & .05 & 0 & 0 & 0 \\ .8 & .15 & .05 & 0 & 0 & 0 \\ .8 & .15 & .05 & 0 & 0 & 0 \end{bmatrix}$$

$$\hat{\alpha} = (I - \beta \overline{P})^{-1} \overline{\gamma}$$

Therefore

$$\hat{\alpha} = \begin{pmatrix} 19.699 \\ 24.857 \\ 30.000 \\ 20.699 \\ 20.699 \\ 20.699 \end{pmatrix}$$

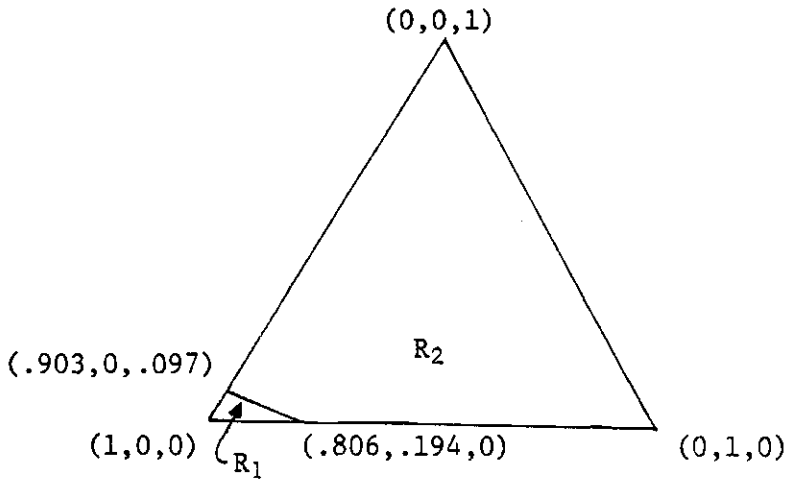
Thus,

$$\hat{\alpha}_1 = \begin{pmatrix} 19.699 \\ 24.857 \\ 30.000 \end{pmatrix} ; \delta_1 = 2$$

and

$$\hat{\alpha}_2 = \begin{pmatrix} 20.699 \\ 20.699 \\ 20.699 \end{pmatrix} ; \delta_2 = 1$$

Graphically, the solution is



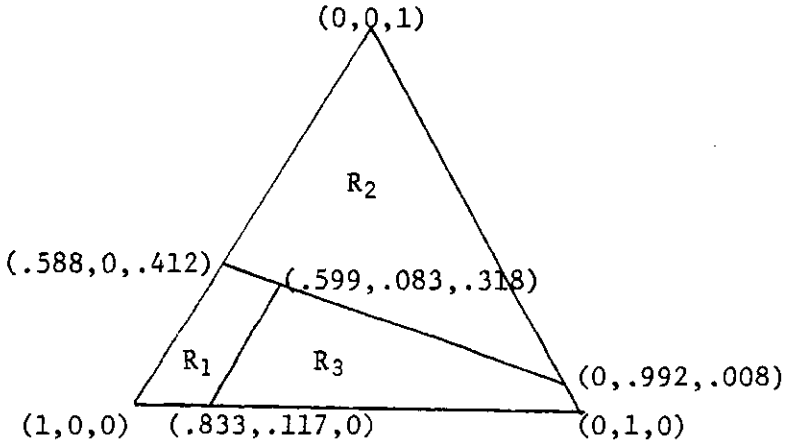
Policy Improvement

To find an improved policy, apply the One-pass Algorithm to the above set of $\hat{\alpha}$-vectors. The resulting new set of α-vectors is

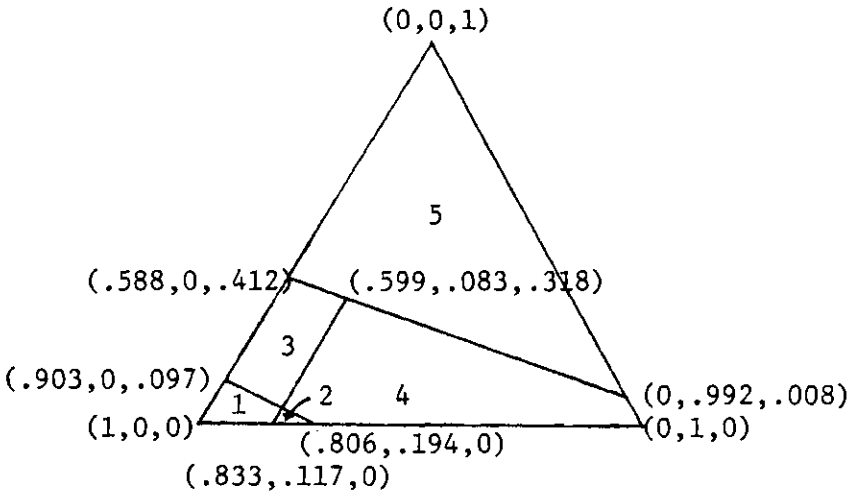
$$\alpha_1 = \begin{pmatrix} 19.200 \\ 21.985 \\ 21.629 \end{pmatrix} \qquad \alpha_2 = \begin{pmatrix} 20.200 \\ 20.200 \\ 20.200 \end{pmatrix} \qquad \alpha_3 = \begin{pmatrix} 19.439 \\ 20.189 \\ 21.629 \end{pmatrix}$$

$$\delta_1 = 2 \qquad \delta_2 = 1 \qquad \delta_3 = 2$$

Graphically:



To check for the largest difference between the two policies, find the regions, through a search procedure, which are the result of the intersection of the regions from the separate policies.



Then check for the maximum error within each region. For this problem, the maximum error is 1.261 and occurs at the point (.903,0,.097). This is too large an error so the steps of the algorithm are repeated.

Value Determination

The policy under consideration is the most recently found set of α-vectors. The mapping for this policy is

	π	$T(\pi\|\theta_1,\delta)$	$\hat{v}(j,\theta_1)$	$T(\pi\|\theta_2,\delta)$	$\hat{v}(j,\theta_2)$
R_1	(1,0,0)	(.877,.123,0)	3	(.593,.222,.185)	3
R_2	(0,.833,.167)	(.877,.123,0)	3	(.593,.222,.185)	3
R_3	(.597,.084,.319)	(.802,.198,0)	3	(.182,.120,.698)	2

$$\bar{\gamma} = \begin{pmatrix} .81 \\ 1.56 \\ 3.00 \\ 1.81 \\ 1.81 \\ 1.81 \\ .81 \\ 1.56 \\ 3.00 \end{pmatrix}$$

$$\bar{P} = \begin{bmatrix} 0 & 0 & P^2 \\ 0 & 0 & P^1 \\ 0 & P^2R_2^2 & P^2R_1^2 \end{bmatrix}$$

The set of $\hat{\alpha}$-vectors is

$$\hat{\alpha}_1 = \begin{pmatrix} 11.625 \\ 13.612 \\ 15.926 \end{pmatrix} \qquad \hat{\alpha}_2 = \begin{pmatrix} 12.625 \\ 12.625 \\ 12.625 \end{pmatrix} \qquad \hat{\alpha}_3 = \begin{pmatrix} 11.658 \\ 13.149 \\ 14.363 \end{pmatrix}$$

$$\delta_1 = 2 \qquad \delta_2 = 1 \qquad \delta_3 = 2$$

Policy Improvement

Applying the One-pass Algorithm to the above set of vectors returns an identical set of vectors (to three decimal places). Therefore there is no further improvement and the above set of α-vectors is the optimal policy.

Implementation of the Algorithm

The sample problems of the preceding sections were simplified in allowing for only two cost observations. This is not realistic in that typically a large number of cost observations is possible, approximating a continuous distribution over a wide interval. To illustrate the implementation of the algorithm, consider the following hypothetical example of a cost variance investigation problem for a manufacturing plant. Cost outputs from an operation in the plant are received on a weekly basis and management decisions related to the operation are made on the basis of a weekly cost report. Assume that the company has been in existence for a sufficient amount of time to have acquired satisfactory sample data for determining the parameters necessary for the cost variance investigation model.

The company has identified four distinct states of the operation, (1) excellent, (2) moderately good, (3) inefficient, and (4) unprofitable. The problem facing management each week is to decide whether the operation is in state 1, and if not, then whether it is profitable to take action to find the cause, or to continue operations unchanged at least for another week.

Consider the following weekly cost report received in one producing department of the plant.

Actual Cost	Standard Cost	Variance
$7625	$7450	$175

Assume that the distribution of the amount of the variance in each of

the four states is constant from week to week, even though the standard cost will vary with the level of activity. In an actual setting, the state-conditional distributions of cost variances would be derived from historical cost data. For purposes of the hypothetical example, these distributions are assumed to be normal.

Since this cost variance investigation model is equipped to handle only a discrete number of cost observations (whereas the underlying distribution is continuous), the possible cost variances are separated into intervals. Intervals of $200 have been arbitrarily chosen for this example. Table 4 shows these intervals and the related state dependent probabilities. Assume that for this example, the remaining parameters, given in Table 5, were estimated from historical data.

Table 4. State-dependent Cost Probabilities for Experimental Problem

Amount of Cost Variance	$P\{\theta \mid x_1\}$	$P\{\theta \mid x_2\}$	$P\{\theta \mid x_3\}$	$P\{\theta \mid x_4\}$
(500)	.0228	.0026	.0002	0
(500)-(300)	.0923	.0202	.0024	.0002
(300)-(100)	.2295	.0923	.0202	.0024
(100)-100	.3108	.2295	.0923	.0202
100-300	.2295	.3108	.2295	.0923
300-500	.0923	.2295	.3108	.2295
500-700	.0202	.0923	.2295	.3108
700-900	.0024	.0202	.0923	.2295
900-1100	.0002	.0024	.0202	.0923
1100-1300	0	.0002	.0024	.0202
1300	0	0	.0002	.0026
Mean	0	200	400	600
σ	250	250	250	250

Table 5. Parameters for Experimental Problem

a_1: Investigate

$$P^1 = \begin{bmatrix} .8 & .15 & .05 & 0 \\ .8 & .15 & .05 & 0 \\ .8 & .15 & .05 & 0 \\ .8 & .15 & .05 & 0 \end{bmatrix}$$

$$\gamma^1 = \begin{pmatrix} 350 \\ 350 \\ 350 \\ 350 \end{pmatrix}$$

a_2: Do Not Investigate

$$P^2 = \begin{bmatrix} .8 & .15 & .05 & 0 \\ .15 & .5 & .25 & .1 \\ 0 & .1 & .7 & .2 \\ 0 & 0 & .1 & .9 \end{bmatrix}$$

$$\gamma^2 = \begin{pmatrix} 50 \\ 260 \\ 420 \\ 580 \end{pmatrix}$$

$$C^\circ = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \beta = 0.98 \qquad K = 300$$

In the final period remaining until operations cease, the α-vectors are simply the immediate expected cost values.

$$\alpha_1^1 = \begin{pmatrix} 50 \\ 260 \\ 420 \\ 580 \end{pmatrix} \qquad \alpha_2^1 = \begin{pmatrix} 350 \\ 350 \\ 350 \\ 350 \end{pmatrix}$$

$$\delta_1^1 = 2 \qquad \delta_2^1 = 1$$

With two periods remaining, the α-vectors are

$$\alpha_1^2 = \begin{pmatrix} 148 \\ 554 \\ 846 \\ 1126 \end{pmatrix} \quad \alpha_2^2 = \begin{pmatrix} 148 \\ 552 \\ 841 \\ 1109 \end{pmatrix} \quad \alpha_3^2 = \begin{pmatrix} 149 \\ 546 \\ 826 \\ 1062 \end{pmatrix} \quad \alpha_4^2 = \begin{pmatrix} 154 \\ 540 \\ 802 \\ 997 \end{pmatrix} \quad \alpha_5^2 = \begin{pmatrix} 448 \\ 448 \\ 448 \\ 448 \end{pmatrix} \quad \alpha_6^2 = \begin{pmatrix} 177 \\ 544 \\ 779 \\ 949 \end{pmatrix}$$

$$\delta_1^2 = 2 \quad \delta_2^2 = 2 \quad \delta_3^2 = 2 \quad \delta_4^2 = 2 \quad \delta_5^2 = 1 \quad \delta_6^2 = 2$$

Assume that at the beginning of the period in which the cost variance report of $175 is received, the state of knowledge is given by $\pi = (1,0,0,0)$. No investigation is undertaken. After receiving the cost report, the revised probability vector is $\pi' = (.794,.201,.005,0)$. The revised probability vector is then multiplied by each of the α-vectors, $\{\alpha_1^2,\ldots,\alpha_6^2\}$; the α-vector corresponding to the minimum product identifies the optimal decision. In this case, the minimum of 232 is given by $\pi' \cdot \alpha_3^2, [=(.794,.201,.005,0) \cdot (149,549,826,1062)]$, and with two periods remaining, an investigation is not warranted.

Consider the situation where management has acquired specific additional knowledge about the state of the process which is not reflected in the received cost report. On the basis of this additional information, one may feel justified in revising the posterior probability vector from the last period.

Let the posterior probability vector remain the same as before where $\pi' = (.794,.201,.005,0)$. Assume that the information management has received prompts a revision of the probability vector to $\pi'' = (.425,$

.150,.250,.175). As before, to determine the optimal decision with two periods remaining, this newly revised probability vector is multiplied by each of the α-vectors, $\{\alpha_1^2,\ldots,\alpha_6^2\}$. The minimum product is 448 and is given by $\pi'' \cdot \alpha_5^2$. This signals a need for an investigation.

Extensions to the Basic Model

To this point, it is assumed that an investigation always reveals the state of the process and is successful in correcting any problems. A complete investigation, however, may be so costly that it is rarely undertaken. Another less costly alternative, such as an exploratory investigation, might be considered. An exploratory investigation is defined as one which will always be successful. In other words, the probability of successfully detecting the cause of being in state i is h_i, with $h_i < 1$. Its cost, K', however, is significantly less than the cost, K, of a complete investigation.

To incorporate this into the model for the investigation of cost variances, the only additional information necessary is the probability transition matrix for this decision alternative. The transition matrix is found by multiplying the following matrix H by the transition matrix for the do-nothing alternative.

$$H = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ h_2 & 1-h_2 & 0 & \cdots & 0 \\ h_3 & 0 & 1-h_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ h_{N_s} & 0 & 0 & \cdots & 1-h_{N_s} \end{bmatrix}$$

Therefore at the beginning of each period of operation, the decision set consists of a do-nothing action, a full investigation, and an exploratory investigation. When a full investigation is made, the state and its cause will be found and the system is reset to state 1 through corrective action. If an exploratory investigation takes place, however, it may not be successful. The system will be reset to state 1 if success is achieved; otherwise operations continue unchanged.

Consider the example given in the first section of this chapter, with the exploratory investigation included as an additional alternative. Let

$$H = \begin{bmatrix} 1 & 0 & 0 \\ .4 & .6 & 0 \\ .6 & 0 & .4 \end{bmatrix}$$

The parameters associated with the third alternative, the exploratory investigation, are shown below.

$$P^3 = \begin{bmatrix} .8 & .15 & .05 \\ .32 & .54 & .14 \\ .48 & .09 & .43 \end{bmatrix} \qquad \gamma^3 = \begin{pmatrix} 1.010 \\ 1.460 \\ 1.886 \end{pmatrix} \qquad K' = .2$$

With three periods remaining, the α-vectors are calculated to be

$$\alpha_1^3 \qquad \alpha_2^3 \qquad \alpha_3^3 \qquad \alpha_4^3 \qquad \alpha_5^3$$

$$\begin{pmatrix} 2.599 \\ 4.185 \\ 5.796 \end{pmatrix} \quad \begin{pmatrix} 2.799 \\ 3.750 \\ 4.077 \end{pmatrix} \quad \begin{pmatrix} 2.898 \\ 3.753 \\ 3.995 \end{pmatrix} \quad \begin{pmatrix} 3.599 \\ 3.599 \\ 3.599 \end{pmatrix} \quad \begin{pmatrix} 2.888 \\ 3.705 \\ 4.131 \end{pmatrix}$$

$$\delta_1^3 = 2 \qquad \delta_2^3 = 3 \qquad \delta_3^3 = 3 \qquad \delta_4^3 = 1 \qquad \delta_5^3 = 3$$

Compare these with the set of α-vectors for three periods remaining in the example having only two decision alternatives, calculated in the finite example in the first section.

$$\alpha_1^3 \qquad \alpha_2^3 \qquad \alpha_3^3 \qquad \alpha_4^3$$

$$\begin{pmatrix} 2.713 \\ 4.673 \\ 7.166 \end{pmatrix} \quad \begin{pmatrix} 2.738 \\ 4.234 \\ 5.465 \end{pmatrix} \quad \begin{pmatrix} 3.713 \\ 3.713 \\ 3.713 \end{pmatrix} \quad \begin{pmatrix} 2.787 \\ 4.173 \\ 5.465 \end{pmatrix}$$

$$\delta_1^3 = 2 \qquad \delta_2^3 = 2 \qquad \delta_3^3 = 1 \qquad \delta_4^3 = 2$$

Notice that when an exploratory investigation is available, the maximum expected cost is 3.599. This is a reduction in the maximum expected cost of 3.713 when no exploratory investigation is available. In this case, the addition of the third alternative leads to only a 3% reduction of the maximum expected cost. This reduction, which is known in advance, may not be significant enough to justify maintaining the exploratory investigation as a viable alternative. The likelihood of conducting an exploratory investigation increases as the h_i's increase and as K' decreases.

A more general treatment of the exploratory investigation is given by Kaplan [14] who discusses briefly the situation where the amount of money to be spent on the investigation is a continuous variable, z. The decision is, therefore, two-fold: first, whether to investigate and second, the amount to spend on the investigation. The probability of detecting the cause of being in an out-of-control state i is a function, $h_i(z)$, of the amount of money spent on the investigation, where $h_i(0) = 0$ and $\lim_{z\to\infty} h_i(z) = 1$, and $h_i(z)$ is non-decreasing.

This expanded or 'continuous' exploratory investigation can easily be integrated into the present model by choosing several discrete values of z to replace the underlying continuous distribution. The modeling as an exploratory investigation then proceeds, except that there are as many decision alternatives as values of z, with $z = 0$ corresponding to the do-nothing alternative.

Budget Constraints on Investigative Activities

Since the amount budgeted for managerial control activity usually limits the number of investigations that can be performed, it may be necessary for the firm to select that set of possible investigations which will minimize total expected cost. As long as the budget period is the same as the cost reporting period, the following integer program which selects the optimal set of investigations is applicable. Since this formulation is a one-period integer program, its solution on a period-by-period basis would only roughly approximate the solution to a complete multi-period dynamic programming formulation of this problem.

$$\min_{i=1,\ldots,s}\left\{ Y_i^1 \min_{j_1:\delta_{j_1}=1} \pi\alpha_{j_1}^i + Y_i^2 \min_{j_2:\delta_{j_2}=2} \pi\alpha_{j_2}^i + Y_i^3 \min_{j_3:\delta_{j_3}=3} \pi\alpha_{j_3}^i \right.$$

$$\text{s.t.} \quad \sum_{i=1}^{s} (Y_i^1 K_i + Y_i^3 K_i') \leq M$$

$$Y_i^1 + Y_i^2 + Y_i^3 = 1$$

$$Y_i^1,\ Y_i^2,\ Y_i^3 = 0 \text{ or } 1$$

where s is the number of operating segments,

M is the budget constraint,

π is the state of knowledge,

K_i is the cost of a full investigation for operation i,

K_i' is the cost of an exploratory investigation for operation i,

α_j^i is the j^{th} α-vector for operation i.

If the value of Y_i^1 is 1, then a full investigation takes place for the i-th operating segment. A value of 1 for Y_i^3 indicates that an exploratory investigation takes place and $Y_i^2 = 1$ indicates no investigation of operation i. The first constraint is the budget constraint and the second and third constraints assure that one and only one decision is made concerning operation i.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

Cost accounting procedures are useful in helping management analyze and control operations. When a reported cost differs from the corresponding standard cost, it is necessary to decide whether the cost variance is significant in the sense of justifying an investigation of the underlying process. To assess this significance, many procedures are available, ranging from rules of thumb, to control charts, to the approach outlined in this thesis.

This work has contributed to cost variance investigation analysis in several ways. The approach employed in this thesis is a method which combines and extends features from the existing work in this field. Previous research is basically limited to cost variance investigation problems in which the underlying process is assumed to be in only one of two possible states, in control or out of control. The present research handles problems characterized by a larger number of states with relative ease. Of importance is the relatively small increase in the amount of computer time as the number of underlying states increases. For example, computer time needed for two different problems each for 10 time periods, each having two cost observation values and two decision alternatives, one with four states, the other with three states, was approximately two seconds for each.

In addition, previous research which was capable of generating

values over an infinite horizon did so only by repeated iterations of the finite time method. The infinite horizon algorithm discussed here does not handle problems in that fashion; computer time is not a function of the number of iterations needed until a steady state situation is encountered. For example, in the three-state problem given in the first section of Chapter V, the computer time using the finite horizon method until steady state values were found was eight seconds, whereas running the same set of data using the algorithm for the infinite horizon, the run time was only 0.4 seconds. Since most firms are ongoing concerns and cost variance problems often arise within the context of an unknown termination time, the infinite-horizon algorithm would be recommended.

It has also been shown in this paper that these two algorithms allow additional considerations to be incorporated into the model with ease, thus increasing flexibility. Extensions to the basic cost variance investigation problem that fit this model include allowing for (1) transitions to occur at the end of the period following the receipt of the cost report [13], (2) a delay in time, say, one time period, from the beginning of an investigation to the conclusion of the correction [14], and (3) the additional decision of not preparing a cost report at all during a time period [11]. Another possible extension involves recognition that there may be multiple causes for being out-of-control. To model this extension, one would simply increase the number of underlying states where each state of the process is a particular combination of being in-control or out-of-control with respect to each of the causes.

Computability has generally not been a problem in the currently available models for cost variance investigation. For this model the

algorithms presented in Chapters III and IV have been thoroughly discussed and their computability has been demonstrated. In Chapter V, the implementation of the algorithms is illustrated both through simple numerical exercises and a more complete experimental problem. In addition, Sondik's infinite horizon algorithm has been simplified and the computer program for both algorithms made available. A completely documented listing of the Fortran program for the finite horizon is found in Appendix D and for the infinite horizon in Appendix E.

Data requirements are quite often the cause of a model being discarded and something else used in its place. Even though the approach presented here requires a larger quantity of data than more primitive methods, the data and parameter requirements are of a more fundamental nature. The parameters can be estimated using data provided by the historical record of the operating system. The procedure for estimating the transition probabilities developed in Appendix A is a case in point. Less judgement and experience are required here than, for example, in a model using the variable labeled L in Dyckman's treatment which requires, in effect, an advance knowledge of when investigations will occur in the future.

It has been shown that the approach in this thesis is flexible, accommodating a large number of variations. However, there are other extensions to the cost variance investigation problem to which the relevance of this model is yet to be demonstrated. As one illustration, consider again the additional alternative of an exploratory investigation. An important question is whether the model utilizes all the information it generates. Other articles that have dealt with this matter

indicate that after an exploratory investigation is performed, if the cause for being in a less than desirable state has not been detected, the state probabilities should immediately be revised. Then, before operations resume, the decision of whether to conduct a full investigation should be made based on revised probabilities. There are times when this is a valid question. However, since it does not seem to fit readily into this model, an area for future study is to ascertain the conditions for which the alternative of a full investigation following an unsuccessful exploratory investigation would never be chosen.

In this context, there are two properties of the probability state space that always exist. They are (1) that the revised probability for state 1 following an unsuccessful exploratory investigation never decreases; that is, $\pi_1' \geq \pi_1$, and (2) that the investigation region is convex. The result of (2) is proved in Appendix C.

It would be interesting to know if, for example, the existence of these two properties along with some restrictions related to the cost data would lead to the class of problems mentioned above, for which a full investigation is never chosen following an unsuccessful exploratory investigation.

Another area for future research is to study the implications of restrictions placed on the transition probabilities. Consider, for example, the determination of optimal replacement rules in a completely observable Markov process setting. Derman [4] has shown that when certain monotonicity-preserving conditions are imposed on the transition probabilities, the optimal replacement rule is to replace if and only if the observed state is one of the states i,i+1,...,L for some i, the

states being ordered from 0 to L with state 0 the most desirable.

In cost variance investigation problems, it seems that, if the same conditions are imposed on the transition probabilities, optimal decision rules could be easily characterized. In other words, it may be possible to separate the region in which investigation decisions are optimal from the do-nothing regions by means of a hyperplane. When this is possible, computational requirements would be drastically reduced as only the separating hyperplane would need to be determined.

In conclusion, the approach to the investigation of cost variances developed in this thesis has many advantages over other methods proposed in the literature. Nevertheless, more complex problem situations continue to exist and can be the subject of future study.

APPENDIX A

THE ESTIMATION OF TRANSITION PROBABILITIES

One set of the required parameters of the model is the set of probabilities in the state transition matrix. These values are seldom known with certainty and must be estimated, usually from past records of the process.

When available, historical transition frequencies are the best means of estimating transition probabilities [15,16]. That is, if a historical record of the state at each time period is available, P_{ij} is estimated by the number of transitions from i to j divided by the number of visits to i. When historical records do not contain transitions but only proportions of time in various states, the methods of Bryant [3] are applicable.

In the cost-variance investigation analysis situation, however, neither of these kinds of data are normally available, because the true state is known only when an investigation is made. Thus the estimation methods must be modified to take into account that the historical record is of the state of knowledge, π, not the underlying state.

For an absorbing two-state model, the only parameter which needs to be estimated is g, where $1 - g$ is the probability of a transition from in-control to out-of-control. Dyckman [7] has provided a method of estimating the mean time between visits to the out-of-control state, whose reciprocal is an estimate of the transition probability.

For this method, consider a series of n past cost observations, $\{\theta_1,\theta_2,\ldots,\theta_n\}$ received between two successive perfect investigations which revealed an out-of-control situation. Given the record of the control decision chosen at the beginning of each time period of the series of cost observations under consideration, the state of knowledge, $\pi(t)$, at each time t, $0 < t \leq n$, can be determined by Bayesian methods where

$$\pi(t) = [\pi_1(t),\pi_2(t)]$$

If an investigation took place at some time $n_1 < n$ where an in-control situation was found, then $\pi(t) = [1,0]$ for all $t \leq n_1$.

Define Q_k as the probability that the process made a transition to the out-of-control state during period k, $1 \leq k \leq n$.

$$Q_k = \pi_1(1)\pi_1(2)\ldots\pi_2(k)\ldots\pi_2(n)$$

Let S_k be the conditional probability that the process made a transition to the out-of-control state during period k, given that an out-of-control state existed at time n. Then

$$S_k = \frac{Q_k}{\sum_{\ell=1}^{n} Q_\ell}$$

Define E(T) as the expected time for the process to move to the out-of-control state for this series of cost observations. Then

$$E(T) = \sum_{k=1}^{n} k\, S_k = \frac{\sum_{k=1}^{n} k\,\theta_k}{\sum_{\ell=1}^{n} Q_\ell}$$

Given E(T) for each of several series of cost observations, an average $\hat{E}(T)$ can be obtained, where $\hat{E}(T)$ is the estimate of the mean passage time to an out-of-control state. The estimate, $1 - \hat{g}$, is the reciprocal of $\hat{E}(T)$.

The development of the parameters for a general probability matrix is more difficult. The following discussion, which is an extension of Dyckman's derivations, is a means for estimating the transition probabilities for processes having more than two states.

Consider a series of past cost observations, $\{\theta_1,\theta_2,\ldots,\theta_n\}$, received between any two successive perfect investigations of a process having N_s states. At the beginning of period 1, the period when θ_1 was reported, the process was known to be in state 1 for an investigation and correction had just occurred. Following the receipt of θ_n, an investigation took place which revealed that the process was in state j. There are $(N_s)^{n-1}$ possible events that could lead to state j at the end of period n. Let E_i be one of the $(N_s)^{n-1}$ possible events. The probability of event E_i is given by

$$Pr\{E_i\} = \pi_{y_1}(1)\ \pi_{y_2}(2)\ \ldots\ \pi_j(n)$$

where y_m is the state of the process at time m for event E_i.

Define Z as the probability that the process is in state j at the

end of period n.

$$Z = \sum_{i=1}^{(N_s)^{n-1}} Pr\{E_i\}$$

Let W_i be the conditional probability of event E_i occurring given state j in period n.

$$W_i = \frac{Pr\{E_i\}}{Z}$$

The element, p_{rs}, in the transition matrix, is the probability of a transition in the process from state r to state s. In estimating p_{rs}, let C_i be the number of times a transition from state r to state s occurs in event E_i. Let T_{ih} be the number of periods for the h-th transition from state r to state s for event E_i. The expected passage time from r to s for the series of cost observations is given by

$$E\big(T(r\to s)\big) = \sum_{i=1}^{(N_s)^{n-1}} \sum_{h=1}^{C_i} T_{ih} \frac{W_i}{\sum_{\ell=1}^{(N_s)^{n-1}} C_\ell W_\ell}$$

where $\sum_{\ell=1}^{(N_s)^{n-1}} C_\ell W_\ell$ is the expected value of the total number of changes from r to s in the n time periods.

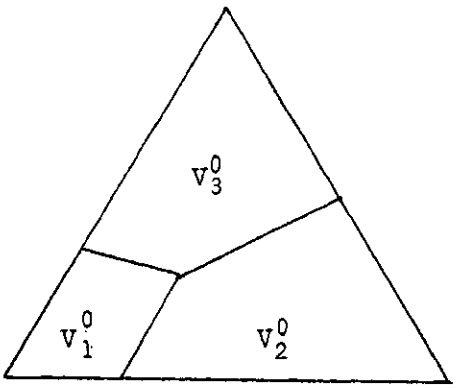
The mean passage time from r to s, $\hat{E}\big(T(r\to s)\big)$, is obtained from the average of $E\big(T(r\to s)\big)$ for several series of cost observations. Thus

$$\hat{p}_{rs} = \frac{1}{\hat{E}\big(T(r\to s)\big)}$$

APPENDIX B

CONSTRUCTION OF THE PARTITION

To determine if a policy is finitely transient, a refined partition for the policy rule δ^∞ must be constructed. For problems having only $N_s = 2$ underlying states, the determination of the partition can be readily expressed graphically; for $N_s \geq 3$, the partition, $V^0 = \{V_1^0, \ldots, V_{M_0}^0\}$, induced by a given stationary policy δ^∞ is first constructed. This partition is usually well-defined by a set of α-vectors, where each V_j^0 is associated with α_j. If $\pi \in V_j^0$, then $\pi\alpha_j = \min \pi\alpha^k$, for all k. Associated with each α_j is a control decision $\delta(\pi) = \delta$. For example,



$\pi \in V_1^0, \quad \delta_1 = 1$

$\pi \in V_2^0, \quad \delta_2 = 2$

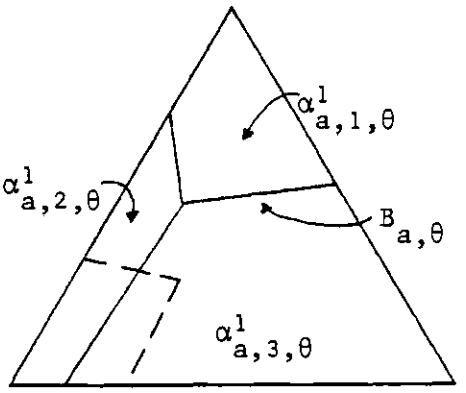
$\pi \in V_3^0, \quad \delta_3 = 2$

The partition V^1 is next formed. Its boundaries are the sets D^1 and D_δ, where D_δ is the smallest closed set in Π for which the policy is discontinuous and

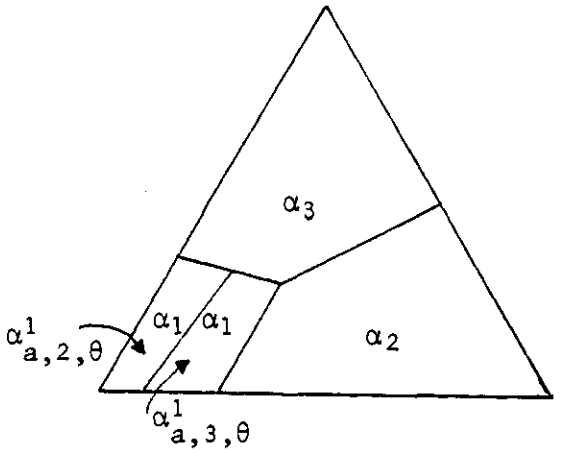
$$D^1 = [\pi : T(\pi | \theta, \delta) \in D_\delta]$$

To see how D^1 is formed, consider the transition of the set Π for some θ using control alternative $a = 1$ as illustrated below. This illustration is similar to that above in that each region is represented by an α-vector, denoted by $\alpha^1_{a,j,\theta}$ where

$$\alpha^1_{a,j,\theta} = P^a R_\theta \alpha_j$$



The boundaries of the above regions are denoted by $B_{a,\theta}$ and are used in forming D^1. Since $a = 1$, the control alternative for region V^0_1, a partial construction of D^1 is the part of $B_{a,\theta}$ which lies in region V^0_1 from the first illustration. The dashed line above indicates the position of V^0_1 relative to $B_{a,\theta}$. Therefore a partial construction of V^1 is as follows:



The procedure is repeated for each θ and for each region V_j^0 to obtain a complete formation of V^1.

Algebraically, the complete construction of V^1 involves finding all vectors of the form $\alpha^1_{a,j,\theta} = P^a R_\theta \alpha_j$ for each $a \in A$, for each θ, and for each j. Each set $V_j^1 \in V^1$ is defined by a set of $N_\theta + 1$ vectors, $\{\alpha^1_j, \alpha^1_{j,\theta_1}, \alpha^1_{j,\theta_2}, \ldots, \alpha^1_{j,\theta_{N_\theta}}\}$ where $\pi \in V_j^1$ if

$$\pi\, \alpha^1_j = \min_k \pi\, \alpha_k \qquad \text{for all } k$$

and for $a = \delta_j$, if

$$\pi\, \alpha^1_{j,\theta_i} = \min_\ell \pi\, \alpha^1_{a,\ell,\theta_i} \qquad \begin{matrix} \text{for all } \ell \\ \text{for all } i \end{matrix}$$

The procedure for defining all the regions is a search technique similar to that used in the One-pass Algorithm in Chapter III. Begin with $V_j^1 = V_1^1$ and initial point $\pi_0 = (1,0,\ldots,0)$. Calculate α_1^1 for π_0 where $\pi_0 \alpha_1^1 = \min_k \pi_0\, \alpha_k$. For $a = \delta(\pi_0)$, find all vectors of the form

$$\alpha^1_{a,\ell,\theta} = P^a R_\theta\, \alpha_\ell \qquad \begin{matrix} \text{for all } \ell \\ \text{for all } \theta \end{matrix}$$

For each θ_i determine α^1_{1,θ_i} such that

$$\pi_0\, \alpha^1_{1,\theta_i} = \pi_0\, \alpha^1_{a,j,\theta_i} = \min_\ell \pi_0\, \alpha^1_{a,\ell,\theta}$$

This completely defines V_1^1.

To find adjacent regions, set up a linear programming problem of

the form

$$\max: \quad f_i$$

$$\text{s.t.} \quad \pi\alpha^1_1 - \pi\alpha_k \leq 0 \qquad \text{for all } k \neq 1 \qquad \text{(a)}$$

$$\pi\alpha^1_{1,\theta_i} - \pi\alpha^1_{a,\ell,\theta_i} \leq 0 \qquad \text{for all } \ell \text{ except for } \alpha^1_{1,\theta_i} = \alpha^1_{a,\ell,\theta_i} \text{ for all } i \qquad \text{(b)}$$

$$\sum \pi_i = 0$$

$$\pi_i \geq 0$$

where each constraint in (b) must be used as the objective function f_i. If f_i is equal to 0 at optimality, then a new region is found. The new region is identified by the same $N_\theta + 1$ vectors as the region being searched except that α_{1,θ_i} becomes α_{a,ℓ,θ_i}. This continues until all the regions are searched. If the number of regions in V^1 is the same as in V^0, then the policy is finitely transient and the partition is complete. If the number of regions in V^1 is greater than in V^0, then V^2 is constructed as follows. Begin with $V^2_j = V^2_1$, $\pi_0 = (1,0,\ldots,0)$ and $a = \delta(\pi_0)$. Identify the region by a set of $2N_\theta + 1$ vectors, $\{\alpha^1_1, \alpha^1_{1,\theta_1}, \ldots \alpha^1_{1,\theta_{N_\theta}}, \alpha^2_{1,\theta_1}, \ldots \alpha^2_{1,\theta_{N_\theta}}\}$ where α^2_{1,θ_i} is defined by $\pi_0\alpha^2_{1,\theta_i} = \min_{k,\theta_\ell} \pi_0 P^a R_{\theta_i} \alpha^1_{k,\theta_\ell}$.

To search adjacent regions, the linear programming problem is established as follows:

$$\max: \quad f_i$$

$$\text{s.t.} \qquad \pi(\alpha_1 - \alpha_k) \leq 0 \qquad \text{for all } k \neq 1 \qquad \text{(a)}$$

$$\pi \alpha^1_{1,\theta_i} - \pi P^a R_{\theta_i} \alpha_k \leq 0 \qquad \text{for all } k \text{ except for } \alpha^1_{1,\theta_i} = P^a R_{\theta_i} \alpha_k \text{ for all } i \qquad \text{(b)}$$

$$\pi \alpha^2_{1,\theta_i} - \pi P^a R_{\theta_i} \alpha^1_{k,\theta_i} \leq 0 \qquad \text{for all } k \text{ for all } \theta \text{ except for } \alpha^2_{1,\theta_i} = P^a R_{\theta_i} \alpha^1_{k,\theta_i} \text{ for all } i \qquad \text{(c)}$$

$$\sum_i \pi_i = 1$$

$$\pi_i \geq 0$$

Each constraint in (c) is in turn used as an objective function. If at optimality, $f_i = 0$, then a new region has been found whose identifying set of vectors is the same as the region being searched except that α^2_{1,θ_i} becomes $P^a R_{\theta_i} \alpha^1_{k,\theta_\ell}$.

The construction of a refined partition continues in like fashion until it is found that the policy is finitely transient.

APPENDIX C

PROOF OF THE CONVEXITY OF THE 'INVESTIGATE' REGION

Consider any two points π_1 and π_2 in the investigate region. Thus

$$C^n(\pi_1) = \pi_1 \gamma^{a_1} + \beta \sum_{\theta} \min_{k} \pi_1 P^{a_1} R_\theta \alpha_k^{n-1} \quad ; k=1,\ldots,S^{(n-1)}$$

and

$$C^n(\pi_2) = \pi_2 \gamma^{a_1} + \beta \sum_{\theta} \min_{k} \pi_2 P^{a_1} R_\theta \alpha_k^{n-1} \quad ; k=1,\ldots,S^{(n-1)}$$

where a_1 is the 'investigate' alternative.

Let

$$\pi = \lambda \pi_1 + (1-\lambda) \pi_2$$

where

$$0 < \lambda \leq 1$$

Since the rows of P^{a_1} are all identical,

$$\pi_1 P^{a_1} = \pi_2 P^{a_1} = \pi P^{a_1}$$

This implies that

$$d(\pi_1,\theta,a_1) = d(\pi_2,\theta,a_1) = d(\pi,\theta,a_1)$$

Therefore

$$C^n(\pi_1) = \pi_1 \gamma^{a_1} + \pi_1 \beta \sum_\theta P^{a_1} R_\theta \alpha^{n-1}_{d(\pi,\theta,a_1)}$$

and

$$C^n(\pi_2) = \pi_2 \gamma^{a_1} + \pi_2 \beta \sum_\theta P^{a_1} R_\theta \alpha^{n-1}_{d(\pi,\theta,a_1)}$$

Let

$$C^n(\pi_1) = \pi_1 \alpha^*$$

and

$$C^n(\pi_2) = \pi_2 \alpha^*$$

where

$$\alpha^* = \gamma^{a_1} + \beta \sum_\theta P^{a_1} R_\theta \alpha^{n-1}_{d(\pi,\theta,a_1)}$$

Thus, α^* is one of the α-vectors in the total set of α-vectors and

$$C^n(\pi) \leq \pi \alpha^* \qquad (1)$$

At the same time

$$C^n(\pi) \geq \lambda C^n(\pi_1) + (1-\lambda) C^n(\pi_2)$$

from the fact that the cost function is piecewise linear and concave.

Therefore,

$$C^n(\pi) \geq \lambda \pi_1 \alpha^* + (1-\lambda) \pi_2 \alpha^*$$

$$= \pi \alpha^* \qquad (2)$$

Together, (1) and (2) imply that

$$C^n(\pi) = \pi \alpha^*$$

and π belongs to the same region as π_1 and π_2. Therefore, the region is convex.

APPENDIX D

FORTRAN PROGRAM FOR THE PARTIALLY OBSERVABLE MARKOV PROCESS OVER A FINITE HORIZON

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION P(4,4,3),R(4,15,3),GAMMA(4,3),ALFOLD(4,60),ALFNEW(4,60)
      DIMENSION IOPT(60)
C
C
C     ************************************************************
C
C     THE FOLLOWING IS THE COMPUTER PROGRAM OF THE ALGORITHM FOR THE PARTIALLY
C     OBSERVABLE MARKOV PROCESS OVER A FINITE HORIZON DEVELOPED IN
C          EDWARD J. SONDIK,'THE OPTIMAL CONTROL OF PARTIALLY
C          OBSERVABLE MARKOV PROCESSES,'PH.D. DISSERTATION, DEPARTMENT OF
C          ENGINEERING-ECONOMIC SYSTEMS, STANFORD UNIVERSITY, STANFORD,
C          CALIFORNIA, JUNE 1971.
C     SEE ALSO,
C          R.D.SMALLWOOD AND E.J.SONDIK,'THE OPTIMAL CONTROL OF PARTIALLY
C          OBSERVABLE MARKOV PROCESSES OVER A FINITE HORIZON,' OPERATIONS
C          RESEARCH, 21 (1973),1071-1088.
C
C     ************************************************************
C
C
C     THE FOLLOWING IS A DESCRIPTION OF ALL ARRAYS AND MAJOR
C     VARIABLES USED IN THIS PROGRAM.
C
C     ARRAYS..............
C
C     P=ARRAY OF STATE TRANSITION PROBABILITIES OF SIZE (I X I X K)
C         WHERE I IS THE NUMBER OF STATES AND K IS THE NUMBER OF DECISION
C         ALTERNATIVES.
C     R=ARRAY OF COST PROBABILITIES OF SIZE (I X J X K) WHERE I IS
C         THE NUMBER OF STATES, J IS THE NUMBER OF COST OBSERVATIONS
C         AND K IS THE NUMBER OF DECISION ALTERNATIVES.
C     GAMMA=ARRAY OF IMMEDIATE EXPECTED COSTS OF SIZE (I X K) WHERE
C         I IS THE NUMBER OF STATES AND K IS THE NUMBER OF DECISION
C         ALTERNATIVES.
C     ALFOLD=ARRAY OF PREVIOUS PERIOD ALPHA-VECTORS OF SIZE (I X M)
C         WHERE I IS THE NUMBER OF STATES AND M IS THE ESTIMATED
C         NUMBER OF REGIONS IN THE PROBABILITY STATE SPACE.
C     ALFNEW=ARRAY OF PRESENT PERIOD ALPHA-VECTORS, SAME SIZE AS
C         ALFOLD ARRAY.
C     IOPT=ARRAY RETURNED FROM SUBROUTINE ONEPAS(ONE-PASS),INDICA-
C         TING OPTIMAL DECISION ALTERNATIVE FOR CORRESPONDING ALFNEW.
C         ARRAY IS OF SIZE M, WHERE M IS THE ESTIMATED NUMBER OF
C         REGIONS IN THE PROBABILITY STATE SPACE.
C     VECTOR=ARRAY OF SIZE I WHERE I IS THE NUMBER OF STATES.  IN
C         ONEPAS AND COSTPI, VECTOR IS USED IN COMPUTATION OF PRODUCTS
C         OF ARRAYS.
C     ARRAY=ARRAY OF COEFFICIENTS OF CONSTRAINTS AND OBJECTIVE
C         FUNCTIONS DEVELOPED IN SEARCH PROCEDURE.  RIGHT HAND SIDE IS
C         IN LOCATION 'IB' OF EACH ROW.
C     ISERCH=ARRAY DESIGNATING WHETHER OR NOT A REGION HAS BEEN SEARCHED
C         FOR ADJACENT REGIONS.
C                   IF ISERCH=0, THE REGION HAS BEEN SEARCHED.
C                   IF ISERCH=1, THE REGION HAS NOT BEEN SEARCHED.
C     IDES=ARRAY USED IN SUBROUTINE ONEPAS TO INDICATE THE MAPPING
C         OF THE PAIRS (DECISION ALTERNATIVE,COST OBSERVATION)
C         ONTO THE SET OF PREVIOUS PERIOD ALPHA-VECTORS.  THE ARRAY
C         IS OF SIZE N, WHERE N IS THE NUMBER OF COST OBSERVATIONS TIMES
C         THE NUMBER OF ALTERNATIVES, WHERE THE COST OBSERVATION INDEX
C         VARIES MOST RAPIDLY.
C     IND=ARRAY FOR STORAGE OF MAPPINGS FOR ALL REGIONS FOUND IN THE
C         PRESENT PERIOD.
C     ALFJA=ARRAY OF ALPHA-VECTOR VALUES FOR EACH ALTERNATIVE.
C     ALFPR=ARRAY OF VECTOR VALUES FORMED BY THE MINIMUM OVER THE PREVIOUS
```

```
C         PERIOD ALPHA-VECTORS OF THE PRODUCT OF P, FOR ALTERNATIVE
C         'A', TIMES DIAGONALIZED R FOR ALTERNATIVE 'A' AND COST OBSER-
C         VATION THETA, TIMES THE PREVIOUS PERIOD ALPHA-VECTOR.
C      CBASIC=ARRAY OF VALUES OF STATE VARIABLES FOR CORRESPONDING
C         OBJECTIVE FUNCTION IN LINEAR PROGRAM AT OPTIMALITY.
C      JND=ARRAY INDICATING STATUS OF EACH OBJECTIVE FUNCTION IN LINEAR
C         PROGRAM.
C                   IF JND=0, THEN OPTIMALITY HAS NOT BEEN REACHED.
C                   IF JND>0, THEN OPTIMALITY HAS BEEN REACHED.
C      IBASIC=ARRAY INDICATING BASIC VARIABLES.
C
C      MAJOR VARIABLES ..............
C
C      NSTATE=NUMBER OF STATES.
C      NTHETA=NUMBER OF POSSIBLE COST OBSERVATIONS.
C      NALTER=NUMBER OF DECISION ALTERNATIVES.
C      NALPHA=NUMBER OF ALPHA-VECTORS IN THE PREVIOUS PERIOD.
C      NTERM=NUMBER OF TOTAL REMAINING TIME PERIODS.
C      BETA=PRESENT VALUE DISCOUNT FACTOR.
C      MCOUNT=NUMBER OF ALPHA-VECTORS IN THE PRESENT PERIOD.
C      NUM=NUMBER OF OBJECTIVE FUNCTIONS IN THE LINEAR PROGRAM.
C      NUM1=POSITION OF THE CONSTRAINT THAT THE SUM OF THE PROBABILITIES
C         TOTAL TO ONE.
C      IB=POSITION OF RIGHT HAND SIDE OF LINEAR PROGRAM.
C      NUMCON=TOTAL NUMBER OF LINES IN LINEAR PROGRAM.
C      ISLAK1=POSITION OF FIRST SLACK VARIABLE.
C      ISLAK2=POSITION OF LAST SLACK VARIABLE.
C      NROW=PIVOT ROW
C      ICOL=PIVOT COLUMN
C      ICON=OBJECTIVE FUNCTION ROW PRESENTLY BEING OPTIMIZED.
C      IROW=INDEX OF BASIC VARIABLE.
C
C
C      ************************************************************
C
C
C      NOTE:  THE PARAMETERS IA AND ID CORRESPOND TO ESTIMATED
C          DIMENSIONS FOR VARIOUS ARRAYS.  THROUGHOUT THE PROGRAM THE ARRAYS
C          ARE TESTED FOR OVER-INDEXING.  THE PROGRAM WILL ABORT IF OVER-
C          INDEXING OCCURS AND A CORRESPONDING MESSAGE WILL BE RECEIVED.
C          WHEN INCREASING IA, CHANGE THE VALUE OF IA AND THE DIMENSIONS
C          OF THE FOLLOWING ARRAYS:  ARRAY(IA,IA),CBASIC(I,IA),JND(IA),
C          IBASIC(IA).  WHEN INCREASING ID, CHANGE THE VALUE OF ID
C          AND THE DIMENSIONS OF THE FOLLOWING ARRAYS:  ALFOLD(I,IA),
C          ALFNEW(I,IA),IOPT(IA),ISERCH(IA),IND(N,IA).
C
C
C      THE ONLY PURPOSE OF THE MAIN PROGRAM IS TO INCREMENT THE TIME
C      PERIODS AND TO WRITE THE VALUES OF THE ALPHA-VECTORS AND THEIR
C      CORRESPONDING OPTIMAL DECISION ALTERNATIVE FOR EACH TIME PERIOD.
C
       READ(5,*)NSTATE,NTHETA,NALTER,NALPHA,NTERM,BETA
       READ(5,*)(((P(M,N,I),N=1,NSTATE),M=1,NSTATE),I=1,NALTER)
       READ(5,*)(((R(M,K,I),K=1,NTHETA),M=1,NSTATE),I=1,NALTER)
       READ(5,*)((GAMMA(M,I),M=1,NSTATE),I=1,NALTER)
       READ(5,*) ((ALFOLD(I,J),J=1,NALPHA),I=1,NSTATE)
       DO 1 N=1,NTERM
       CALL ONEPAS(P,R,GAMMA,ALFOLD,NSTATE,NTHETA,NALTER,NALPHA,BETA,
      1 ALFNEW,IOPT,MCOUNT)
       PRINT *,' THE NUMBER OF PERIODS REMAINING UNTIL TERMINATION IS ',N
       WRITE(6,2)(IOPT(J),J=1,MCOUNT)
2      FORMAT(13I10)
       DO 5 I = 1,NSTATE
       WRITE(6,4) (ALFNEW(I,J),J=1,MCOUNT)
4      FORMAT(13F10.3)
5      CONTINUE
```

```
      WRITE(6,3)
3     FORMAT(1X,1H*,3X,1H*,3X,1H*,3X,1H*)
      NALPHA=MCOUNT
      DO 6 I=1,NALPHA
      DO 6 M=1,NSTATE
6     ALFOLD(M,I)=ALFNEW(M,I)
1     CONTINUE
      STOP
      END
C
C
      SUBROUTINE ONEPAS(P,R,GAMMA,ALFOLD,NSTATE,NTHETA,NALTER,NALPHA,
     2 BETA,ALFNEW,IOPT,MCOUNT)
      DIMENSION P(4,4,3),R(4,15,3),GAMMA(4,3),ALFOLD(4,60)
      DIMENSION ALFNEW(4,60),IOPT(60),VECTOR(4),ARRAY(200,200)
      DIMENSION ISERCH(60),IDES(45),IND(45,60),ALFJA(4,3)
      DIMENSION ALFPR(4,45),CBASIC(4,200)
C
C     THE PURPOSE OF THE SUBROUTINE ONEPAS(ONE-PASS) IS TO CALCULATE
C     THE ALPHA-VECTORS FOR THE PRESENT PERIOD.
C
      IA=200
      ID=60
      CALL COSTPI(P,R,GAMMA,ALFOLD,NSTATE,NTHETA,NALTER,
     1 NALPHA,BETA,IOPT(1),IDES)
C-----INITIALIZE FOR REGION NUMBER ONE.
      MM=NTHETA*NALTER
      DO 1 M=1,MM
1     IND(M,1)=IDES(M)
      MCOUNT=1
      ISERCH(1)=1
C
C-----PICK A REGION TO BE SEARCHED.
C
26    DO 2 I=1,MCOUNT
      IF(ISERCH(I).EQ.0)GO TO 2
      ISERCH(I)=0
      GO TO 3
2     CONTINUE
      GO TO 53
3     JALPHA=I
      NUM=0
C     FOR GIVEN REGION, ASSIGN MAPPING VALUES TO IDES.
      MM=NTHETA*NALTER
      DO 12 M=1,MM
12    IDES(M)=IND(M,I)
C-----FOR GIVEN MAPPING, COMPUTE VALUES OF ALFJA FOR EACH ALTERNATIVE.
      DO 4 I=1,NALTER
      DO 5 M=1,NSTATE
5     ALFJA(M,I)=0.
C-----FOR GIVEN MAPPING, COMPUTE VALUES OF ALFPR FOR EACH PAIR,
C-----  (COST OBSERVATION,DECISION ALTERNATIVE).
      DO 6 K=1,NTHETA
      NN=(K-1)*NALTER+I
      KK=IDES(NN)
      II=(I-1)*NTHETA+K
      DO 7 M=1,NSTATE
      ALFPR(M,II)=0.
7     VECTOR(M)=R(M,K,I)*ALFOLD(M,KK)
      DO 6 M=1,NSTATE
      DO 6 N=1,NSTATE
6     ALFPR(M,II)=ALFPR(M,II)+P(M,N,I)*VECTOR(N)
      DO 4 M=1,NSTATE
      DO 8 K=1,NTHETA
      II=(I-1)*NTHETA+K
8     ALFJA(M,I)=ALFJA(M,I)+ALFPR(M,II)
```

```
4       ALFJA(M,I)=ALFJA(M,I)*BETA+GAMMA(M,I)
C
C-----SET UP LINEAR PROGRAM.
C
C-----FORMATION OF COEFFICIENTS FOR NON-SLACK VARIABLES.
        DO 9 I=1,NALTER
        IF(I.EQ.IOPT(JALPHA))GO TO 9
        NUM=NUM+1
        IF(NUM.GT.IA)GO TO 601
        DO 10 N=1,NSTATE
10      ARRAY(NUM,N)=ALFJA(N,IOPT(JALPHA))-ALFJA(N,I)
9       CONTINUE
        DO 11 I=1,NALTER
        DO 11 K=1,NTHETA
        NN=(K-1)*NALTER+I
        KK=IDES(NN)
        II=(I-1)*NTHETA+K
        DO 13 L=1,NALPHA
        IF(L.EQ.KK)GO TO 13
        NUM=NUM+1
        IF(NUM.GT.IA)GO TO 601
        DO 14 N=1,NSTATE
        ARRAY(NUM,N)=0.
14      VECTOR(N)=R(N,K,I)*ALFOLD(N,L)
        DO 15 M=1,NSTATE
        DO 15 N=1,NSTATE
15      ARRAY(NUM,M)=ARRAY(NUM,M)+P(M,N,I)*VECTOR(N)
        DO 16 N=1,NSTATE
16      ARRAY(NUM,N)=ALFPR(N,II)-ARRAY(NUM,N)
13      CONTINUE
11      CONTINUE
        IF(JALPHA.NE.1)GO TO 17
        DO 18 N=1,NSTATE
18      ALFNEW(N,1)=ALFJA(N,IOPT(1))
C-----COMPLETE FORMATION OF LINEAR PROGRAM.
17      NUM1=NUM+1
        IF(NUM1.GT.IA)GO TO 601
        IB=NSTATE+NUM1
        IF(IB.GT.IA)GO TO 601
        NUMCON=NUM1+NUM
        IF(NUMCON.GT.IA)GO TO 601
        ISLAK1=NSTATE+1
        ISLAK2=IB-1
        DO 19 I=1,NUM
        DO 19 J=ISLAK1,IB
19      ARRAY(I,J)=0.
        DO 20 I=1,NSTATE
20      ARRAY(NUM1,I)=1.
        DO 21 I=ISLAK1,ISLAK2
21      ARRAY(NUM1,I)=0.
        ARRAY(NUM1,IB)=1.
        DO 22 I=1,NUM
        DO 22 J=1,IB
22      ARRAY(I+NUM1,J)=ARRAY(I,J)
        DO 23 I=1,NUM
        ARRAY(I+NUM1,IB)=0.
23      ARRAY(I+NUM1,I+NSTATE)=1.
        CALL LINEAR(ARRAY,NUM,NUMCON,IB,ISLAK1,ISLAK2,
     1  NSTATE,CBASIC)
C
C-----CHECK FOR BINDING CONSTRAINTS.
C
        DO 24 M=1,NUM
        J2=0
        IF(ARRAY(M,IB).GT.1.E-6)GO TO 24
C-----IDENTIFY THE VARIABLES USED IN THE FORMATION OF THE CONSTRAINTS.
```

```
      DO 25 I=1,NALTER
       IF(I.EQ.IOPT(JALPHA))GO TO 25
       J2=J2+1
       IF(M.EQ.J2)GO TO 34
25     CONTINUE
       DO 35 I=1,NALTER
       DO 35 K=1,NTHETA
       NN=(K-1)*NALTER+I
       KK=IDES(NN)
       II=(I-1)*NTHETA+K
       DO 36 L=1,NALPHA
       IF(L.EQ.KK)GO TO 36
       J2=J2+1
       IF(M.EQ.J2)GO TO 37
36     CONTINUE
35     CONTINUE
C
C-----BINDING CONSTRAINT IS OF TYPE 1:  FORMED FROM PRESENT PERIOD
C-----  ALPHA-VECTORS.
C
34     NN=NTHETA*NALTER
C-----DETERMINE IF NEW-FOUND REGION IS ALREADY IN LIST OF REGIONS.
       DO 38 J=1,MCOUNT
       DO 39 N=1,NSTATE
       IF(ABS(ALFJA(N,I)-ALFNEW(N,J)).GT.1.E-6)GO TO 38
39     CONTINUE
       DO 40 N=1,NN
       IF(IDES(N).NE.IND(N,J))GO TO 38
40     CONTINUE
       GO TO 24
38     CONTINUE
C-----INCREMENT NUMBER OF REGIONS IN LIST.
       MCOUNT=MCOUNT+1
       IF(MCOUNT.GT.ID)GO TO 604
       ISERCH(MCOUNT)=1
       IOPT(MCOUNT)=I
       DO 41 N=1,NSTATE
41     ALFNEW(N,MCOUNT)=ALFJA(N,I)
       DO 42 N=1,NN
42     IND(N,MCOUNT)=IDES(N)
       GO TO 24
C
C-----BINDING CONSTRAINT IS OF TYPE 2:  FORMED FROM PREVIOUS PERIOD
C-----  ALPHA-VECTORS.
C
37     J2=NTHETA*NALTER
       IF(MCOUNT .EQ.ID)GO TO 604
C-----DETERMINE IF REGION IS ALREADY IN LIST OF REGIONS.
       DO 48 N=1,J2
48     IND(N,MCOUNT+1)=IDES(N)
       IND(NN,MCOUNT+1)=L
       IF(I.NE.IOPT(JALPHA)) GO TO 43
C-----DECISION ATERNATIVE OF NEW REGION IS SAME AS REGION BEING SEARCHED.
       DO 44 N=1,NSTATE
       ALFNEW(N,MCOUNT+1)=0.
44     VECTOR(N)=R(N,K,I)*ALFOLD(N,L)
       DO 45 J=1,NSTATE
       DO 45 N=1,NSTATE
45     ALFNEW(J,MCOUNT+1)=ALFNEW(J,MCOUNT+1)+P(J,N,I)*VECTOR(N)
       DO 52 J=1,NSTATE
       ALFNEW(J,MCOUNT+1)=ALFJA(J,I)+BETA*(ALFNEW(J,MCOUNT+1)
      1 -ALFPR(J,II))
52     CONTINUE
51     DO 46 J=1,MCOUNT
       DO 47 N=1,NSTATE
       IF(ABS(ALFNEW(N,MCOUNT+1)-ALFNEW(N,J)).GT.1.E-6)GO TO 46
```

```
47    CONTINUE
      DO 49 N=1,J2
      IF(IND(N,MCOUNT+1).NE.IND(N,J))GO TO 46
49    CONTINUE
      GO TO 24
46    CONTINUE
C-----INCREMENT NUMBER OF REGIONS IN LIST.
      MCOUNT=MCOUNT+1
      ISERCH(MCOUNT)=1
      IOPT(MCOUNT)=IOPT(JALPHA)
      GO TO 24
C-----DECISION ALTERNATIVE OF NEW REGION IS DIFFERENT FROM REGION
C-----  BEING SEARCHED.
43    DO 50 N=1,NSTATE
50    ALFNEW(N,MCOUNT+1)=ALFNEW(N,JALPHA)
      GO TO 51
24    CONTINUE
      GO TO 26
C-----FORM NEW LIST OF REGIONS, DIFFERING WITH RESPECT TO ONLY DECISION
C-----  ALTERNATIVE AND ALPHA-VECTOR.
53    MM=MCOUNT
      MCOUNT=1
      DO 54 N=2,MM
      DO 55 M=1,MCOUNT
      DO 56 I=1,NSTATE
      IF(ABS(ALFNEW(I,N)-ALFNEW(I,M)).GT.1.E-6)GO TO 55
56    CONTINUE
      GO TO 54
55    CONTINUE
      MCOUNT=MCOUNT+1
      DO 57 I=1,NSTATE
57    ALFNEW(I,MCOUNT)=ALFNEW(I,N)
      IOPT(MCOUNT)=IOPT(N)
54    CONTINUE
      RETURN
601   PRINT *,'INCREASE IA'
      STOP
604   PRINT *,'INCREASE ID'
      STOP
      END
C
C
      SUBROUTINE LINEAR(ARRAY,NUM,NUMCON,IB,ISLAK1,ISLAK2,NSTATE,
     1 CBASIC)
      DIMENSION ARRAY(200,200),CBASIC(4,200),IBASIC(200),JND(200)
C
C     THE PURPOSE OF THE SUBROUTINE IS TO FIND THE OPTIMAL VALUES
C     FOR THE LINEAR PROGRAM.
C
C-----INITIALIZE.
      LINES=NUMCON-NUM
      NUM1=NUM+1
      DO 1 I=1,NUM
1     JND(I)=0
      NROW = NUM1
C
C-----USING THE EXPANDED DUAL SIMPLEX ALGORITHM.
C
C-----STEP 1:  BASIS-SEEKING STEP. ROW NUM1 HAS NO BASIC VARIABLE.
C-----  PIVOT ON ROW NUM1 AND COLUMN ICOL.  COLUMN ICOL IS CHOSEN SO
C-----  THAT RATIO A(1,I)/A(NUM1,I) IS SMALLEST.
      RATIO = -1000000.
      DO 20 I=1,NSTATE
      B = ARRAY(1,I)/ARRAY(NROW,I)
      IF(B .LE. RATIO) GO TO 20
      RATIO = B
```

```
      ICOL = I
20    CONTINUE
      IBASIC(1)=ICOL
      II = NROW - 1
      DO 16 I=1,II
      B = ARRAY(I,ICOL)
      DO 16 M=1,IB
16    ARRAY(I,M)=ARRAY(I,M)-B*ARRAY(NROW,M)
      II = NROW + 1
      DO 17 I=II,NUMCON
      B = ARRAY(I,ICOL)
      DO 17 M=1,IB
17    ARRAY(I,M)=ARRAY(I,M)-B*ARRAY(NROW,M)
      DO 18 I=2,LINES
18    IBASIC(I) = NSTATE + I-1
C-----STEP 2: FEASIBILITY-SEEKING STEP.  PIVOT ON ROW NROW, COLUMN ICOL.
C-----  NROW IS ROW HAVING THE MOST NEGATIVE RIGHT HAND SIDE.  ICOL
C-----  IS COLUMN HAVING NEGATIVE ELEMENT IN PIVOT ROW WITH MOST
C-----  FAVORABLE RATIO A(1,I)/A(NROW,I)
66    B = -1.E-6
      DO 65 I=NUM1,NUMCON
      IF(ARRAY(I,IB) .GE. B) GO TO 65
      B=ARRAY(I,IB)
      NROW=I
65    CONTINUE
      IF(B.GE.-1.E-6)GO TO 22
      J1=0
      RATIO=0.
      DO 60 I=1,ISLAK2
      IF(ARRAY(NROW,I).LE.-1.E-6)GO TO 61
60    CONTINUE
      DO 59 I=1,NUM
59    ARRAY(I,IB)=1000000.
      RETURN
61    DO 67 I=1,ISLAK2
      IF(ARRAY(NROW,I).GT.-1.*1.E-6)GO TO 67
      IF(ARRAY(1,I).LT.1.E-6)GO TO 67
      IF(ARRAY(1,I)/ARRAY(NROW,I).GE.RATIO)GO TO 67
      RATIO = ARRAY(1,I)/ARRAY(NROW,I)
      J1=1
      ICOL=I
67    CONTINUE
      IF(J1.EQ.1)GO TO 68
      RATIO=1000000.
      DO 69 I=1,ISLAK2
      IF(ARRAY(NROW,I).GT.-1.*1.E-6)GO TO 69
      IF(ARRAY(1,I)/ARRAY(NROW,I).GE.RATIO)GO TO 69
      RATIO=ARRAY(1,I)/ARRAY(NROW,I)
      ICOL=I
69    CONTINUE
68    IROW=NROW-NUM
      IBASIC(IROW)=ICOL
      B=ARRAY(NROW,ICOL)
      DO 70 M=1,IB
70    ARRAY(NROW,M)=ARRAY(NROW,M)/B
      II=NROW-1
      DO 71 I=1,II
      B=ARRAY(I,ICOL)
      DO 71 M=1,IB
71    ARRAY(I,M)=ARRAY(I,M)-B*ARRAY(NROW,M)
      II=NROW+1
      IF(II.GT.NUMCON)GO TO 15
      DO 72 I=II,NUMCON
      B=ARRAY(I,ICOL)
      DO 72 M=1,IB
72    ARRAY(I,M)=ARRAY(I,M)-B*ARRAY(NROW,M)
```

```
15    GO TO 66
C-----STEP 3:  OPTIMALITY-SEEKING STEP.  PIVOT RULES ARE IDENTICAL
C-----  TO SIMPLEX ITERATION.
22    DO 19 I=NUM1,NUMCON
      IF(ARRAY(I,IB).LT.0.)ARRAY(I,IB)=0.
19    CONTINUE
      J1=0
C-----AN OBJECTIVE FUNCTION, PREVIOUSLY OPTIMIZED, IS NOT AFFECTED
C-----  BY ADDITIONAL PIVOTS.
      DO 23 M=1,NUM
      IF(JND(M).EQ.1)GO TO 23
      J2=0
      DO 24 N=1,ISLAK2
      IF(ARRAY(M,N).LT.1.E-6)GO TO 24
      J2=1
      IF(J1.EQ.1)GO TO 25
      ICON=M
      J1=1
      GO TO 25
24    CONTINUE
25    IF(J2.EQ.1)GO TO 23
      JND(M)=1
      DO 34 I=1,NSTATE
34    CBASIC(I,M)=0.
      DO 35 I=1,LINES
      II=IBASIC(I)
      IF(II.GT.NSTATE)GO TO 35
      CBASIC(II,M)=ARRAY(I+NUM,IB)
35    CONTINUE
23    CONTINUE
      IF(J1.EQ.0)GO TO 99
      AOPT=1.E-6
      DO 26 N=1,ISLAK2
      IF(ARRAY(ICON,N).LT.AOPT)GO TO 26
      AOPT=ARRAY(ICON,N)
      ICOL=N
26    CONTINUE
      RATIO=1000000.
      DO 29 N=NUM1,NUMCON
      IF(ARRAY(N,ICOL).LT.1.E-6) GO TO 29
      IF(ARRAY(N,IB)/ARRAY(N,ICOL).GE.RATIO)GO TO 29
      IROW=N-NUM
      RATIO=ARRAY(N,IB)/ARRAY(N,ICOL)
29    CONTINUE
      IBASIC(IROW)=ICOL
      NROW=NUM+IROW
      B=ARRAY(NROW,ICOL)
      DO 30 M=1,IB
30    ARRAY(NROW,M)=ARRAY(NROW,M)/B
      II=NROW-1
      DO 31 I=1,II
      IF(I.GT.NUM)GO TO 62
      IF(JND(I).EQ.1)GO TO 31
62    B=ARRAY(I,ICOL)
      DO 33 M=1,IB
33    ARRAY(I,M)=ARRAY(I,M)-B*ARRAY(NROW,M)
31    CONTINUE
      II=NROW+1
      IF(II.GT.NUMCON)GO TO 28
      DO 32 I=II,NUMCON
      B=ARRAY(I,ICOL)
      DO 32 M=1,IB
32    ARRAY(I,M)=ARRAY(I,M)-B*ARRAY(NROW,M)
28    GO TO 22
99    RETURN
      END
```

```
C
      SUBROUTINE COSTPI(P,R,GAMMA,ALFOLD,NSTATE,NTHETA,NALTER,NALPHA,
     1 BETA,IOPT,IDES)
      DIMENSION PI(4),P(4,4,3),R(4,15,3),GAMMA(4,3),ALFOLD(4,60)
      DIMENSION IDES(45),VECTOR(4)
C
C     THIS SUBROUTINE RECEIVES THE INITIAL DATA EACH TIME PERIOD AND
C     RETURNS THE OPTIMAL ALTERNATIVE AND THE MAPPING ONTO THE
C     PREVIOUS PERIOD ALPHA-VECTORS AT THE POINT (1,0,...,0).
C
      PI(1) = 1.
      DO 8 M = 2,NSTATE
8     PI(M) = 0.
      C = 1000000.
      DO 1 I=1,NALTER
      TSUM = 0.
      DO 2 K = 1,NTHETA
      SMIN = 1000000.
      DO 3 M=1,NSTATE
      VECTOR(M) = 0.
      DO 4 N=1,NSTATE
4     VECTOR(M) = VECTOR(M)+PI(N)*P(N,M,I)
3     VECTOR(M) = VECTOR(M)*R(M,K,I)
      DO 5 N=1,NALPHA
      SUM = 0.
      DO 6 M= 1,NSTATE
6     SUM=SUM +VECTOR (M)*ALFOLD(M,N)
      IF(SMIN .LE. SUM) GO TO 5
      SMIN = SUM
      MM=(K-1)*NALTER+I
      IDES(MM) = N
5     CONTINUE
2     TSUM = TSUM + SMIN
      TSUM = BETA*TSUM
      SUM = 0.
      DO 7 N = 1,NSTATE
7     SUM = SUM + PI(N)*GAMMA(N,I)
      SUM = SUM + TSUM
      IF(C .LE. SUM) GO TO 1
      IOPT = I
      C = SUM
1     CONTINUE
      RETURN
      END
```

APPENDIX E

FORTRAN PROGRAM FOR THE PARTIALLY OBSERVABLE MARKOV PROCESS OVER AN INFINITE HORIZON

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION P(4,4,3),R(4,15,3),GAMMA(4,3),ALFOLD(4,60),ALFNEW(4,60)
      DIMENSION IOPT(60),PI(4,60),VECTOR(4),ARRAY(125,125)
      DIMENSION ISERCH(60),IREG(15,60),CBASIC(4,125)
      DIMENSION GAMMA2(65),PHAT(65,65),PHAT2(65,65),WKAREA(65)
C
C
C     ************************************************************
C
C     THE FOLLOWING IS THE COMPUTER PROGRAM OF THE ALGORITHM FOR THE PARTIALLY
C     OBSERVABLE MARKOV PROCESS OVER AN INFINITE HORIZON DEVELOPED IN
C          EDWARD J. SONDIK,'THE OPTIMAL CONTROL OF PARTIALLY
C          OBSERVABLE MARKOV PROCESSES,'PH.D. DISSERTATION, DEPARTMENT OF
C          ENGINEERING-ECONOMIC SYSTEMS, STANFORD UNIVERSITY, STANFORD,
C          CALIFORNIA, JUNE 1971.
C     SEE ALSO,
C          E.J.SONDIK,'THE OPTIMAL CONTROL OF PARTIALLY OBSERVABLE MARKOV
C          PROCESSES OVER THE INFINITE HORIZON:  DISCOUNTED COSTS,' OPERATIONS
C          RESEARCH, 26 (1978), 282-304.
C
C     ************************************************************
C
C
C     THE FOLLOWING IS A DESCRIPTION OF ALL ARRAYS AND MAJOR
C     VARIABLES USED IN THIS PROGRAM.
C
C     ARRAYS..............
C
C     P=ARRAY OF STATE TRANSITION PROBABILITIES OF SIZE (I X I X K)
C         WHERE I IS THE NUMBER OF STATES AND K IS THE NUMBER OF DECISION
C         ALTERNATIVES.
C     R=ARRAY OF COST PROBABILITIES OF SIZE (I X J X K) WHERE I IS
C         THE NUMBER OF STATES, J IS THE NUMBER OF COST OBSERVATIONS
C         AND K IS THE NUMBER OF DECISION ALTERNATIVES.
C     GAMMA=ARRAY OF IMMEDIATE EXPECTED COSTS OF SIZE (I X K) WHERE
C         I IS THE NUMBER OF STATES AND K IS THE NUMBER OF DECISION
C         ALTERNATIVES.
C     ALFOLD=ARRAY OF PREVIOUS PERIOD ALPHA-VECTORS OF SIZE (I X M)
C         WHERE I IS THE NUMBER OF STATES AND M IS THE ESTIMATED
C         NUMBER OF REGIONS IN THE PROBABILITY STATE SPACE.
C     ALFNEW=ARRAY OF PRESENT PERIOD ALPHA-VECTORS, SAME SIZE AS
C         ALFOLD ARRAY.
C     IOPT=ARRAY RETURNED FROM SUBROUTINE ONEPAS(ONE-PASS),INDICA-
C         TING OPTIMAL DECISION ALTERNATIVE FOR CORRESPONDING ALFNEW.
C         ARRAY IS OF SIZE M, WHERE M IS THE ESTIMATED NUMBER OF
C         REGIONS IN THE PROBABILITY STATE SPACE.
C     PI=ARRAY RETURNED FROM SUBROUTINE ONEPAS OF SIZE (I X M) WHERE
C         I IS THE NUMBER OF STATES AND M IS THE ESTIMATED NUMBER OF REGIONS
C         IN THE PROBABILITY STATE SPACE.  THE M TH COLUMN OF PI IS A
C         POINT IN REGION M OF THE PROBABILITY STATE SPACE.  PI IN COSTPI IS
C         DIFFERENT ARRAY OF DIMENSION (I).
C     VECTOR=ARRAY OF SIZE I WHERE I IS THE NUMBER OF STATES.  IN MAIN,
C         VECTOR IS USED IN COMPUTION OF TRANSFORMATION OF PI.  IN
C         ONEPAS AND COSTPI, VECTOR IS USED IN COMPUTATION OF PRODUCTS
C         OF ARRAYS.
C     ARRAY=ARRAY OF COEFFICIENTS OF CONSTRAINTS AND OBJECTIVE
C         FUNCTIONS DEVELOPED IN SEARCH PROCEDURE.  RIGHT HAND SIDE IS
C         IN LOCATION 'IB' OF EACH ROW.
C     ISERCH=ARRAY DESIGNATING WHETHER OR NOT A REGION HAS BEEN SEARCHED
C         FOR ADJACENT REGIONS.
C                   IF ISERCH=0, THE REGION HAS BEEN SEARCHED.
C                   IF ISERCH=1, THE REGION HAS NOT BEEN SEARCHED.
C     IREG=ARRAY OF SIZE (J X M) USED TWO WAYS.  FIRST, STORES
C         REGION OF TRANSFORMED PI FOR EACH THETA IN EACH REGION.
C         ALSO, USED IN COMPARING OLD AND NEW POLICIES.  SEARCH
C         PROCEDURE FINDS ALL REGIONS FORMED BY THE INTERSECTION
```

```
C          OF REGIONS FROM OLD AND NEW POLICIES.  ROW 1 OF IREG DESIGNATES
C          ALPHA-VECTOR FOR OLD POLICY.  ROW 2 OF IREG DESIGNATES ALPHA-
C          VECTOR FOR NEW POLICY.
C      CBASIC=ARRAY OF VALUES OF STATE VARIABLES FOR CORRESPONDING
C          OBJECTIVE FUNCTION IN LINEAR PROGRAM AT OPTIMALITY.
C      GAMMA2=VECTOR OF VECTORS OF SIZE (M TIMES NUMBER OF STATES) FORMED
C          BY COMBINING THE VECTORS FOR IMMEDIATE EXPECTED COST FOR EACH
C          REGION IN THE PROBABILITY STATE SPACE.
C      PHAT (P-HAT) =ARRAY OF (I-BETA) TIMES P(BAR), WHERE P(BAR) IS A
C          MATRIX OF MATRICES FORMED USING THE MAPPING OF THE REGIONS OF
C          THE PROBABILITY STATE SPACE ONTO THE PROBABILITY STATE SPACE.
C      PHAT2=ARRAY RETURNED FROM EXTERNAL SUBROUTINE LINV1F, THE INVERSE OF PHAT.
C      WKAREA=ARTIFICIAL ARRAY REQUIRED FOR EXTERNAL SUBROUTINE LINV1F OF
C          SAME DIMENSION AS PHAT.
C      IDES=ARRAY USED IN SUBROUTINE ONEPAS TO INDICATE THE MAPPING
C          OF THE PAIRS (DECISION ALTERNATIVE,COST OBSERVATION)
C          ONTO THE SET OF PREVIOUS PERIOD ALPHA-VECTORS.  THE ARRAY
C          IS OF SIZE N, WHERE N IS THE NUMBER OF COST OBSERVATIONS TIMES
C          THE NUMBER OF ALTERNATIVES, WHERE THE COST OBSERVATION INDEX
C          VARIES MOST RAPIDLY.
C      IND=ARRAY FOR STORAGE OF MAPPINGS FOR ALL REGIONS FOUND IN THE
C          PRESENT PERIOD.
C      ALFJA=ARRAY OF ALPHA-VECTOR VALUES FOR EACH ALTERNATIVE.
C      ALFPR=ARRAY OF VECTOR VALUES FORMED BY THE MINIMUM OVER THE PREVIOUS
C          PERIOD ALPHA-VECTORS OF THE PRODUCT OF P, FOR ALTERNATIVE
C          'A', TIMES DIAGONALIZED R FOR ALTERNATIVE 'A' AND COST OBSER-
C          VATION THETA, TIMES THE PREVIOUS PERIOD ALPHA-VECTOR.
C      JND=ARRAY INDICATING STATUS OF EACH OBJECTIVE FUNCTION IN LINEAR
C          PROGRAM.
C                   IF JND=0, THEN OPTIMALITY HAS NOT BEEN REACHED.
C                   IF JND>0, THEN OPTIMALITY HAS BEEN REACHED.
C      IBASIC=ARRAY INDICATING BASIC VARIABLES.
C
C      MAJOR VARIABLES ..............
C
C      NSTATE=NUMBER OF STATES.
C      NTHETA=NUMBER OF POSSIBLE COST OBSERVATIONS.
C      NALTER=NUMBER OF DECISION ALTERNATIVES.
C      NALPHA=NUMBER OF ALPHA-VECTORS IN THE PREVIOUS PERIOD.
C      BETA=PRESENT VALUE DISCOUNT FACTOR.
C      ERROR1=MAXIMUM ERROR ALLOWABLE BETWEEN OLD AND NEW POLICIES.
C      NPART=NUMBER OF TIMES ONEPAS IS CALLED FOR ALFOLD BEFORE
C          DETERMINATION OF NEW POLICY.
C      MCOUNT=NUMBER OF ALPHA-VECTORS IN THE PRESENT PERIOD.
C      NUM=NUMBER OF OBJECTIVE FUNCTIONS IN THE LINEAR PROGRAM.
C      NUM1=POSITION OF THE CONSTRAINT THAT THE SUM OF THE PROBABILITIES
C          TOTAL TO ONE.
C      IB=POSITION OF RIGHT HAND SIDE OF LINEAR PROGRAM.
C      NUMCON=TOTAL NUMBER OF LINES IN LINEAR PROGRAM.
C      ISLAK1=POSITION OF FIRST SLACK VARIABLE.
C      ISLAK2=POSITION OF LAST SLACK VARIABLE.
C      NROW=PIVOT ROW
C      ICOL=PIVOT COLUMN
C      ICON=OBJECTIVE FUNCTION ROW PRESENTLY BEING OPTIMIZED.
C      IROW=INDEX OF BASIC VARIABLE.
C
C
C      ************************************************************
C
C
C      NOTE:  THE PARAMETERS IA,ID,AND IE CORRESPOND TO ESTIMATED
C          DIMENSIONS FOR VARIOUS ARRAYS.  THROUGHOUT THE PROGRAM THE ARRAYS
C          ARE TESTED FOR OVER-INDEXING.  THE PROGRAM WILL ABORT IF OVER-
C          INDEXING OCCURS AND A CORRESPONDING MESSAGE WILL BE RECEIVED.
C          WHEN INCREASING IA, CHANGE THE VALUE OF IA AND THE DIMENSIONS
C          OF THE FOLLOWING ARRAYS:  ARRAY(IA,IA),CBASIC(I,IA),JND(IA),
```

```
C          IBASIC(IA).  WHEN INCREASING ID, CHANGE THE VALUE OF ID
C          AND THE DIMENSIONS OF THE FOLLOWING ARRAYS:  ALFOLD(I,ID),
C          ALFNEW(I,ID),IOPT(ID),ISERCH(ID),IND(N,ID).  WHEN INCREASING
C          IE, CHANGE THE VALUE OF IE AND THE DIMENSIONS OF THE FOLLOWING
C          ARRAYS:  GAMMA2(IE),PHAT(IE,IE),PHAT2(IE,IE),WKAREA(IE).
C
C
      READ(5,*)NSTATE,NTHETA,NALTER,NALPHA,BETA,ERROR1,NPART
      READ(5,*)(((P(M,N,I),N=1,NSTATE),M=1,NSTATE),I=1,NALTER)
      READ(5,*)(((R(M,K,I),K=1,NTHETA),M=1,NSTATE),I=1,NALTER)
      READ(5,*)((GAMMA(M,I),M=1,NSTATE),I=1,NALTER)
      READ(5,*) ((ALFOLD(I,J),J=1,NALPHA),I=1,NSTATE)
      IA=125
      ID=60
      IE=65
98    FORMAT(10F10.3)
97    FORMAT(10I4)
C
C-----CALCULATE NEW ALPHA-VECTORS,GIVEN OLD ALPHA-VECTORS.
C
      CALL ONEPAS(P,R,GAMMA,ALFOLD,NSTATE,NTHETA,NALTER,NALPHA,BETA,
     1 ALFNEW,IOPT,MCOUNT,PI)
C-----REPLACE OLD ALPHA-VECTORS WITH NEW ALPHA-VECTORS.
55    PRINT *,'NEW POLICY'
      DO 1 I=1,MCOUNT
      DO 57 M=1,NSTATE
57    ALFOLD(M,I)=ALFNEW(M,I)
      WRITE(6,98)(ALFNEW(M,I),M=1,NSTATE)
1     CONTINUE
      NALPHA=MCOUNT
C-----FOR A POINT IN EACH REGION, CALCULATE THE VALUE OF THE TRANSFOR-
C-----  MATION OF THE POINT.
      DO 2 I=1,NALPHA
      PRINT *,' OPTIMAL DECISION IS ',IOPT(I),' FOR PI = '
      WRITE(6,98)(PI(M,I),M=1,NSTATE)
      PRINT *,'TRANSFORMED VALUES OF PI'
      DO 3 K=1,NTHETA
4     DO 5 M=1,NSTATE
      VECTOR(M)=0.
      DO 6 N=1,NSTATE
6     VECTOR(M)=VECTOR(M)+PI(N,I)*P(N,M,IOPT(I))
5     VECTOR(M)=VECTOR(M)*R(M,K,IOPT(I))
      B=0.
      DO 7 M=1,NSTATE
7     B=B+VECTOR(M)
      IF(B.EQ.0.) B=1.
      DO 8 M=1,NSTATE
8     VECTOR(M)=VECTOR(M)/B
      WRITE(6,98)(VECTOR(M),M=1,NSTATE)
C-----FOR EACH THETA, DETERMINE THE REGION OF THE TRANSFORMED POINT
C-----  IN EACH REGION.
      T=1000000.
      DO 9 J=1,NALPHA
      B=0.
      DO 10 M=1,NSTATE
10    B=B+VECTOR(M)*ALFOLD(M,J)
      IF(B.GT.T)GO TO 9
      IREG(K,I)=J
      T=B
9     CONTINUE
3     CONTINUE
      PRINT *,'REGIONS'
      WRITE(6,*)(IREG(K,I),K=1,NTHETA)
2     CONTINUE
C
C-----CALCULATE THE COST FUNCTION ALPHA-VECTORS.
```

```
C
      IF(NALPHA*NSTATE.GT.IE)GO TO 605
      DO 11 I=1,NALPHA
      DO 11 N=1,NSTATE
      II=(I-1)*NSTATE+N
11    GAMMA2(II)=GAMMA(N,IOPT(I))
      II=NALPHA*NSTATE
      DO 12 I=1,II
      DO 12 J=1,II
12    PHAT(I,J)=0.
      DO 13 I=1,NALPHA
      DO 13 K=1,NTHETA
      DO 14 J=1,NALPHA
      IF(IREG(K,I).NE.J)GO TO 14
      II=(I-1)*NSTATE
      JJ=(J-1)*NSTATE
      DO 15 M=1,NSTATE
      DO 15 N=1,NSTATE
15    PHAT(II+M,JJ+N)=PHAT(II+M,JJ+N)-BETA*P(M,N,IOPT(I))*R(N,K,IOPT(I))
14    CONTINUE
13    CONTINUE
      II=NALPHA*NSTATE
      DO 16 I=1,II
16    PHAT(I,I)=1.+PHAT(I,I)
C-----EXTERNAL SUBROUTINE LINV1F FINDS THE INVERSE OF ARRAY PHAT.
C-----  THE INVERSE IS RETURNED IN PHAT2.  IER IS AN ERROR PARAMETER
C-----  RETURNED FROM LINV1F.
      CALL LINV1F(PHAT,II,IE,PHAT2,3,WKAREA,IER)
      DO 17 I=1,NALPHA
      DO 17 M=1,NSTATE
17    ALFNEW(M,I)=0.
      JJ=NSTATE*NALPHA
      DO 18 I=1,NALPHA
      II=(I-1)*NSTATE
      DO 18 M=1,NSTATE
      DO 18 J=1,JJ
18    ALFNEW(M,I)=ALFNEW(M,I)+PHAT2(II+M,J)*GAMMA2(J)
      PRINT *,'ALPHA-HAT VECTORS'
      DO 61 I=1,NALPHA
      WRITE(6,98)(ALFNEW(M,I),M=1,NSTATE)
61    CONTINUE
      MM=NALPHA
C-----PLACE THE CALCULATED COST FUNCTION ALPHA-VECTORS IN ALFOLD.
      NALPHA=1
      DO 56 I=1,NSTATE
56    ALFOLD(I,1)=ALFNEW(I,1)
      DO 19 N=2,MM
      DO 20 M=1,NALPHA
      DO 21 I=1,NSTATE
      IF(ABS(ALFNEW(I,N)-ALFNEW(I,M)).GT.1.E-6)GO TO 20
21    CONTINUE
      GO TO 19
20    CONTINUE
      NALPHA=NALPHA+1
      DO 22 I=1,NSTATE
22    ALFOLD(I,NALPHA)=ALFNEW(I,N)
19    CONTINUE
      CALL ONEPAS(P,R,GAMMA,ALFOLD,NSTATE,NTHETA,NALTER,NALPHA,BETA,
     1 ALFNEW,IOPT,MCOUNT,PI)
      PRINT *,'IMPROVED POLICY'
      DO 62 I=1,MCOUNT
      WRITE(6,98)(ALFNEW(M,I),M=1,NSTATE)
62    CONTINUE
      PRINT *,'CHECKING ERROR OVER OLD POLICY'
C
C-----FIND ALL REGIONS FROMED BY THE INTERSECTION OF THE REGIONS
```

```
C-----  CORRESPONDING TO THE OLD ALPHA-VECTORS AND THE REGIONS
C-----  CORRESPONDING TO THE NEW ALPHA-VECTORS.
      NN=1
      ISERCH(1)=1
      IREG(1,1)=1
      IREG(2,1)=1
 42   DO 23 I=1,NN
      IF(ISERCH(I).EQ.0)GO TO 23
      ISERCH(I)=0
      GO TO 24
 23   CONTINUE
      GO TO 25
 24   JALPHA=IREG(1,I)
      JJ=IREG(2,I)
      NUM=0
      DO 26 I=1,NALPHA
      IF(I.EQ.JALPHA)GO TO 26
      NUM=NUM+1
      IF(NUM.GT.IA)GO TO 601
      DO 27 M=1,NSTATE
 27   ARRAY(NUM,M)=ALFOLD(M,JALPHA)-ALFOLD(M,I)
 26   CONTINUE
      DO 28 I=1,MCOUNT
      IF(I.EQ.JJ)GO TO 28
      NUM=NUM+1
      IF(NUM.GT.IA)GO TO 601
      DO 29 M=1,NSTATE
 29   ARRAY(NUM,M)=ALFNEW(M,JJ)-ALFNEW(M,I)
 28   CONTINUE
      NUM1=NUM+1
      NUMCON=NUM1
      IF(NUMCON.GT.IA)GO TO 601
      DO 30 M=1,NSTATE
 30   ARRAY(NUMCON,M)=1.
      DO 31 I=1,NUM
      NUMCON=NUMCON+1
      IF(NUMCON.GT.IA)GO TO 601
      DO 31 M=1,NSTATE
 31   ARRAY(NUMCON,M)=ARRAY(I,M)
      ISLAK1=NSTATE+1
      ISLAK2=NSTATE+NUM
      IB=ISLAK2+1
      IF(IB.GT.IA)GO TO 601
      DO 32 I=1,NUMCON
      DO 32 M=ISLAK1,IB
 32   ARRAY(I,M)=0.
      ARRAY(NUM1,IB)=1.
      DO 33 I=1,NUM
 33   ARRAY(NUM1+I,NSTATE+I)=1.
      CALL LINEAR(ARRAY,NUM,NUMCON,IB,ISLAK1,ISLAK2,NSTATE,
     1  CBASIC)
      DO 34 M=1,NUM
      IF(ARRAY(M,IB).GT.1.E-6)GO TO 34
      J2=0
      DO 35 I=1,NALPHA
      IF(I.EQ.JALPHA)GO TO 35
      J2=J2+1
      IF(M.EQ.J2)GO TO 36
 35   CONTINUE
      DO 37 I=1,MCOUNT
      IF(I.EQ.JJ)GO TO 37
      J2=J2+1
      IF(M.EQ.J2)GO TO 39
 37   CONTINUE
 36   IREG(1,NN+1)=I
      IREG(2,NN+1)=JJ
```

```
      GO TO 38
39    IREG(1,NN+1)=JALPHA
      IREG(2,NN+1)=I
38    DO 40 J=1,NN
      DO 41 L=1,2
      IF(IREG(L,J).NE.IREG(L,NN+1))GO TO 40
41    CONTINUE
      GO TO 34
40    CONTINUE
      NN=NN+1
      IF(NN.GT.ID)GO TO 604
      ISERCH(NN)=1
34    CONTINUE
      GO TO 42
25    CONTINUE
C
C-----FOR EACH REGION FOUND IN THE PRECEDING SECTION, FIND THE MAXIMUM
C-----  DIFFERENCE BETWEEN THE TWO POLICIES.
C
      ERROR=0
      NUM=1
      DO 43 I=1,NN
      DO 44 M=1,NSTATE
44    ARRAY(NUM,M)=ALFOLD(M,IREG(1,I))-ALFNEW(M,IREG(2,I))
      NUM1=2
      NUMCON=2
      DO 45 M=1,NSTATE
45    ARRAY(NUM1,M)=1.
      DO 46 J=1,NALPHA
      IF(J.EQ.IREG(1,I))GO TO 46
      NUMCON=NUMCON+1
      DO 47 M=1,NSTATE
47    ARRAY(NUMCON,M)=ALFOLD(M,IREG(1,I))-ALFOLD(M,J)
46    CONTINUE
      DO 48 J=1,MCOUNT
      IF(J.EQ.IREG(2,I))GO TO 48
      NUMCON=NUMCON+1
      DO 49 M=1,NSTATE
49    ARRAY(NUMCON,M)=ALFNEW(M,IREG(2,I))-ALFNEW(M,J)
48    CONTINUE
      ISLAK1=NSTATE+1
      ISLAK2=NSTATE+NUMCON-2
      IB=ISLAK2+1
      DO 50 M=1,NUMCON
      DO 50 N=ISLAK1,ISLAK2
50    ARRAY(M,N)=0.
      ARRAY(NUM1,IB)=1.
      ARRAY(NUM,IB)=0.
      DO 51 N=3,NUMCON
      II=N-2
      ARRAY(N,IB)=-.01
51    ARRAY(N,NSTATE+II)=1.
      CALL LINEAR(ARRAY,NUM,NUMCON,IB,ISLAK1,ISLAK2,NSTATE,
     1  CBASIC)
      ARRAY(NUM,IB)=-ARRAY(NUM,IB)
      IF(ERROR.LT.ARRAY(NUM,IB))ERROR=ARRAY(NUM,IB)
43    CONTINUE
      PRINT *,'ERROR = ',ERROR
      IF(ERROR.LE.ERROR1)GO TO 52
C
C-----CALL ONEPAS NPART TIMES TO FORM A REFINED PARTITION.
C
      N=1
60    IF(N.GT.NPART)GO TO 55
      DO 58 I=1,MCOUNT
      DO 59 M=1,NSTATE
```

```
59      ALFOLD(M,I)=ALFNEW(M,I)
58      CONTINUE
        NALPHA=MCOUNT
        CALL ONEPAS(P,R,GAMMA,ALFOLD,NSTATE,NTHETA,NALTER,NALPHA,
       2 BETA,ALFNEW,IOPT,MCOUNT,PI)
        N=N+1
        GO TO 60
52      DO 54 I=1,MCOUNT
        WRITE(6,*)IOPT(I)
        WRITE(6,98)(ALFNEW(J,I),J=1,NSTATE)
54      CONTINUE
        STOP
601     PRINT *,'INCREASE IA'
        STOP
604     PRINT *,'INCREASE ID'
        STOP
605     PRINT *,'INCREASE IE'
         STOP
        END
C
C
        SUBROUTINE ONEPAS(P,R,GAMMA,ALFOLD,NSTATE,NTHETA,NALTER,NALPHA,
       2 BETA,ALFNEW,IOPT,MCOUNT,PI)
        DIMENSION P(4,4,3),R(4,15,3),GAMMA(4,3),ALFOLD(4,60)
        DIMENSION ALFNEW(4,60),IOPT(60),VECTOR(4),ARRAY(125,125)
        DIMENSION ISERCH(60),IDES(45),IND(45,60),ALFJA(4,3)
        DIMENSION ALFPR(4,45),CBASIC(4,125),PI(4,60)
C
C       THE PURPOSE OF THE SUBROUTINE ONEPAS(ONE-PASS) IS TO CALCULATE
C       THE ALPHA-VECTORS FOR THE PRESENT PERIOD.
C
        IA=125
        ID=60
        CALL COSTPI(P,R,GAMMA,ALFOLD,NSTATE,NTHETA,NALTER,
       1 NALPHA,BETA,IOPT(1),IDES)
C-----INITIALIZE FOR REGION NUMBER ONE.
        PI(1,1)=1.
        DO 27 I=2,NSTATE
27      PI(I,1)=0.
        MM=NTHETA*NALTER
        DO 1 M=1,MM
1       IND(M,1)=IDES(M)
        MCOUNT=1
        ISERCH(1)=1
C
C-----PICK A REGION TO BE SEARCHED.
C
26      DO 2 I=1,MCOUNT
        IF(ISERCH(I).EQ.0)GO TO 2
        ISERCH(I)=0
        GO TO 3
2       CONTINUE
        GO TO 53
3       JALPHA=I
        NUM=0
C-----FOR GIVEN REGION, ASSIGN MAPPING VALUES TO IDES.
        MM=NTHETA*NALTER
        DO 12 M=1,MM
12      IDES(M)=IND(M,I)
C-----FOR GIVEN MAPPING, COMPUTE VALUES OF ALFJA FOR EACH ALTERNATIVE.
        DO 4 I=1,NALTER
        DO 5 M=1,NSTATE
5       ALFJA(M,I)=0.
C-----FOR GIVEN MAPPING, COMPUTE VALUES OF ALFPR FOR EACH PAIR,
C-----  (COST OBSERVATION,DECISION ALTERNATIVE).
        DO 6 K=1,NTHETA
```

```
      NN=(K-1)*NALTER+I
      KK=IDES(NN)
      II=(I-1)*NTHETA+K
      DO 7 M=1,NSTATE
      ALFPR(M,II)=0.
7     VECTOR(M)=R(M,K,I)*ALFOLD(M,KK)
      DO 6 M=1,NSTATE
      DO 6 N=1,NSTATE
6     ALFPR(M,II)=ALFPR(M,II)+P(M,N,I)*VECTOR(N)
      DO 4 M=1,NSTATE
      DO 8 K=1,NTHETA
      II=(I-1)*NTHETA+K
8     ALFJA(M,I)=ALFJA(M,I)+ALFPR(M,II)
4     ALFJA(M,I)=ALFJA(M,I)*BETA+GAMMA(M,I)
C
C-----SET UP LINEAR PROGRAM.
C
C-----FORMATION OF COEFFICIENTS FOR NON-SLACK VARIABLES.
      DO 9 I=1,NALTER
      IF(I.EQ.IOPT(JALPHA))GO TO 9
      NUM=NUM+1
      IF(NUM.GT.IA)GO TO 601
      DO 10 N=1,NSTATE
10    ARRAY(NUM,N)=ALFJA(N,IOPT(JALPHA))-ALFJA(N,I)
9     CONTINUE
      DO 11 I=1,NALTER
      DO 11 K=1,NTHETA
      NN=(K-1)*NALTER+I
      KK=IDES(NN)
      II=(I-1)*NTHETA+K
      DO 13 L=1,NALPHA
      IF(L.EQ.KK)GO TO 13
      NUM=NUM+1
      IF(NUM.GT.IA)GO TO 601
      DO 14 N=1,NSTATE
      ARRAY(NUM,N)=0.
14    VECTOR(N)=R(N,K,I)*ALFOLD(N,L)
      DO 15 M=1,NSTATE
      DO 15 N=1,NSTATE
15    ARRAY(NUM,M)=ARRAY(NUM,M)+P(M,N,I)*VECTOR(N)
      DO 16 N=1,NSTATE
16    ARRAY(NUM,N)=ALFPR(N,II)-ARRAY(NUM,N)
13    CONTINUE
11    CONTINUE
      IF(JALPHA.NE.1)GO TO 17
      DO 18 N=1,NSTATE
18    ALFNEW(N,1)=ALFJA(N,IOPT(1))
C-----COMPLETE FORMATION OF LINEAR PROGRAM.
17    NUM1=NUM+1
      IF(NUM1.GT.IA)GO TO 601
      IB=NSTATE+NUM1
      IF(IB.GT.IA)GO TO 601
      NUMCON=NUM1+NUM
      IF(NUMCON.GT.IA)GO TO 601
      ISLAK1=NSTATE+1
      ISLAK2=IB-1
      DO 19 I=1,NUM
      DO 19 J=ISLAK1,IB
19    ARRAY(I,J)=0.
      DO 20 I=1,NSTATE
20    ARRAY(NUM1,I)=1.
      DO 21 I=ISLAK1,ISLAK2
21    ARRAY(NUM1,I)=0.
      ARRAY(NUM1,IB)=1.
      DO 22 I=1,NUM
      DO 22 J=1,IB
```

```
22    ARRAY(I+NUM1,J)=ARRAY(I,J)
      DO 23 I=1,NUM
      ARRAY(I+NUM1,IB)=.01
C-----ASSIGN .01 TO RIGHT HAND SIDE OF CONSTRAINTS TO ALLOW FOR ROUND-
C-----  OFF ERRORS.
23    ARRAY(I+NUM1,I+NSTATE)=1.
      CALL LINEAR(ARRAY,NUM,NUMCON,IB,ISLAK1,ISLAK2,
     1  NSTATE,CBASIC)
C
C-----CHECK FOR BINDING CONSTRAINTS.
C
      DO 24 M=1,NUM
      J2=0
      IF(ARRAY(M,IB).GT.-.009999)GO TO 24
C-----IDENTIFY THE VARIABLES USED IN THE FORMATION OF THE CONSTRAINTS.
      DO 25 I=1,NALTER
      IF(I.EQ.IOPT(JALPHA))GO TO 25
      J2=J2+1
      IF(M.EQ.J2)GO TO 34
25    CONTINUE
      DO 35 I=1,NALTER
      DO 35 K=1,NTHETA
      NN=(K-1)*NALTER+I
      KK=IDES(NN)
      II=(I-1)*NTHETA+K
      DO 36 L=1,NALPHA
      IF(L.EQ.KK)GO TO 36
      J2=J2+1
      IF(M.EQ.J2)GO TO 37
36    CONTINUE
35    CONTINUE
C
C-----BINDING CONSTRAINT IS OF TYPE 1:  FORMED FROM PRESENT PERIOD
C-----  ALPHA-VECTORS.
C
34    NN=NTHETA*NALTER
C-----DETERMINE IF NEW-FOUND REGION IS ALREADY IN LIST OF REGIONS.
      DO 38 J=1,MCOUNT
      DO 39 N=1,NSTATE
      IF(ABS(ALFJA(N,I)-ALFNEW(N,J)).GT.1.E-6)GO TO 38
39    CONTINUE
      DO 40 N=1,NN
      IF(IDES(N).NE.IND(N,J))GO TO 38
40    CONTINUE
      GO TO 24
38    CONTINUE
C-----INCREMENT NUMBER OF REGIONS IN LIST.
      MCOUNT=MCOUNT+1
      IF(MCOUNT .GT. ID)GO TO 604
      ISERCH(MCOUNT)=1
      IOPT(MCOUNT)=I
      DO 41 N=1,NSTATE
      PI(N,MCOUNT)=CBASIC(N,M)
41    ALFNEW(N,MCOUNT)=ALFJA(N,I)
      DO 42 N=1,NN
42    IND(N,MCOUNT)=IDES(N)
      GO TO 24
C
C-----BINDING CONSTRAINT IS OF TYPE 2:  FORMED FROM PREVIOUS PERIOD
C-----  ALPHA-VECTORS.
C
37    J2=NTHETA*NALTER
      IF(MCOUNT .EQ. ID)GO TO 604
C-----DETERMINE IF REGION IS ALREADY IN LIST OF REGIONS.
      DO 48 N=1,J2
48    IND(N,MCOUNT+1)=IDES(N)
```

```
      IND(NN,MCOUNT+1)=L
      IF(I.NE.IOPT(JALPHA)) GO TO 43
C-----DECISION ATERNATIVE OF NEW REGION IS SAME AS REGION BEING SEARCHED.
      DO 44 N=1,NSTATE
      ALFNEW(N,MCOUNT+1)=0.
44    VECTOR(N)=R(N,K,I)*ALFOLD(N,L)
      DO 45 J=1,NSTATE
      DO 45 N=1,NSTATE
45    ALFNEW(J,MCOUNT+1)=ALFNEW(J,MCOUNT+1)+P(J,N,I)*VECTOR(N)
      DO 52 J=1,NSTATE
      ALFNEW(J,MCOUNT+1)=ALFJA(J,I)+BETA*(ALFNEW(J,MCOUNT+1)
     1 -ALFPR(J,II))
52    CONTINUE
51    DO 46 J=1,MCOUNT
      DO 47 N=1,NSTATE
      IF(ABS(ALFNEW(N,MCOUNT+1)-ALFNEW(N,J)).GT.1.E-6)GO TO 46
47    CONTINUE
      DO 49 N=1,J2
      IF(IND(N,MCOUNT+1).NE.IND(N,J))GO TO 46
49    CONTINUE
      GO TO 24
46    CONTINUE
C-----INCREMENT NUMBER OF REGIONS IN LIST.
      MCOUNT=MCOUNT+1
      ISERCH(MCOUNT)=1
      IOPT(MCOUNT)=IOPT(JALPHA)
      DO 28 N=1,NSTATE
28    PI(N,MCOUNT)=CBASIC(N,M)
      GO TO 24
C-----DECISION ALTERNATIVE OF NEW REGION IS DIFFERENT FROM REGION
C-----  BEING SEARCHED.
43    DO 50 N=1,NSTATE
50    ALFNEW(N,MCOUNT+1)=ALFNEW(N,JALPHA)
      GO TO 51
24    CONTINUE
      GO TO 26
C-----FORM NEW LIST OF REGIONS, DIFFERING WITH RESPECT TO ONLY DECISION
C-----  ALTERNATIVE AND ALPHA-VECTOR.
53    MM=MCOUNT
      MCOUNT=1
      DO 54 N=2,MM
      DO 55 M=1,MCOUNT
      DO 56 I=1,NSTATE
      IF(ABS(ALFNEW(I,N)-ALFNEW(I,M)).GT.1.E-6)GO TO 55
56    CONTINUE
      GO TO 54
55    CONTINUE
      MCOUNT=MCOUNT+1
      DO 57 I=1,NSTATE
      PI(I,MCOUNT)=PI(I,N)
57    ALFNEW(I,MCOUNT)=ALFNEW(I,N)
      IOPT(MCOUNT)=IOPT(N)
54    CONTINUE
      RETURN
601   PRINT *,'INCREASE IA'
      STOP
604   PRINT *,'INCREASE ID'
      STOP
      END
C
C
      SUBROUTINE LINEAR(ARRAY,NUM,NUMCON,IB,ISLAK1,ISLAK2,NSTATE,
     1 CBASIC)
      DIMENSION ARRAY(125,125),CBASIC(4,125),IBASIC(125),JND(125)
C
C     THE PURPOSE OF THE SUBROUTINE IS TO FIND THE OPTIMAL VALUES
```

```
C     FOR THE LINEAR PROGRAM.
C
C-----INITIALIZE.
      LINES=NUMCON-NUM
      NUM1=NUM+1
      DO 1 I=1,NUM
1     JND(I)=0
      NROW = NUM1
C
C-----USING THE EXPANDED DUAL SIMPLEX ALGORITHM.
C
C-----STEP 1:  BASIS-SEEKING STEP. ROW NUM1 HAS NO BASIC VARIABLE.
C-----  PIVOT ON ROW NUM1 AND COLUMN ICOL.  COLUMN ICOL IS CHOSEN SO
C-----  THAT RATIO A(1,I)/A(NUM1,I) IS SMALLEST.
      RATIO = -1000000.
      DO 20 I=1,NSTATE
      B = ARRAY(1,I)/ARRAY(NROW,I)
      IF(B .LE. RATIO) GO TO 20
      RATIO = B
      ICOL = I
20    CONTINUE
      IBASIC(1)=ICOL
      II = NROW - 1
      DO 16 I=1,II
      B = ARRAY(I,ICOL)
      DO 16 M=1,IB
16    ARRAY(I,M)=ARRAY(I,M)-B*ARRAY(NROW,M)
      II = NROW + 1
      DO 17 I=II,NUMCON
      B = ARRAY(I,ICOL)
      DO 17 M=1,IB
17    ARRAY(I,M)=ARRAY(I,M)-B*ARRAY(NROW,M)
      DO 18 I=2,LINES
18    IBASIC(I) = NSTATE + I-1
C-----STEP 2: FEASIBILITY-SEEKING STEP.  PIVOT ON ROW NROW, COLUMN ICOL.
C-----  NROW IS ROW HAVING THE MOST NEGATIVE RIGHT HAND SIDE.  ICOL
C-----  IS COLUMN HAVING NEGATIVE ELEMENT IN PIVOT ROW WITH MOST
C-----  FAVORABLE RATIO A(1,I)/A(NROW,I)
66    B = -1.E-6
      DO 65 I=NUM1,NUMCON
      IF(ARRAY(I,IB) .GE. B) GO TO 65
      B=ARRAY(I,IB)
      NROW=I
65    CONTINUE
      IF(B.GE.-1.E-6)GO TO 22
      J1=0
      RATIO=0.
      DO 60 I=1,ISLAK2
      IF(ARRAY(NROW,I).LE.-1.E-6)GO TO 61
60    CONTINUE
      DO 59 I=1,NUM
59    ARRAY(I,IB)=1000000.
      RETURN
61    DO 67 I=1,ISLAK2
      IF(ARRAY(NROW,I).GT.-1.*1.E-6)GO TO 67
      IF(ARRAY(1,I).LT.1.E-6)GO TO 67
      IF(ARRAY(1,I)/ARRAY(NROW,I).GE.RATIO)GO TO 67
      RATIO = ARRAY(1,I)/ARRAY(NROW,I)
      J1=1
      ICOL=I
67    CONTINUE
      IF(J1.EQ.1)GO TO 68
      RATIO=1000000.
      DO 69 I=1,ISLAK2
      IF(ARRAY(NROW,I).GT.-1.*1.E-6)GO TO 69
      IF(ARRAY(1,I)/ARRAY(NROW,I).GE.RATIO)GO TO 69
```

```
      RATIO=ARRAY(1,I)/ARRAY(NROW,I)
      ICOL=I
69    CONTINUE
68    IROW=NROW-NUM
      IBASIC(IROW)=ICOL
      B=ARRAY(NROW,ICOL)
      DO 70 M=1,IB
70    ARRAY(NROW,M)=ARRAY(NROW,M)/B
      II=NROW-1
      DO 71 I=1,II
      B=ARRAY(I,ICOL)
      DO 71 M=1,IB
71    ARRAY(I,M)=ARRAY(I,M)-B*ARRAY(NROW,M)
      II=NROW+1
      IF(II.GT.NUMCON)GO TO 15
      DO 72 I=II,NUMCON
      B=ARRAY(I,ICOL)
      DO 72 M=1,IB
72    ARRAY(I,M)=ARRAY(I,M)-B*ARRAY(NROW,M)
15    GO TO 66
C-----STEP 3:  OPTIMALITY-SEEKING STEP.  PIVOT RULES ARE IDENTICAL
C-----  TO SIMPLEX ITERATION.
22    DO 19 I=NUM1,NUMCON
      IF(ARRAY(I,IB).LT.0.)ARRAY(I,IB)=0.
19    CONTINUE
      J1=0
C-----AN OBJECTIVE FUNCTION, PREVIOUSLY OPTIMIZED, IS NOT AFFECTED
C-----  BY ADDITIONAL PIVOTS.
      DO 23 M=1,NUM
      IF(JND(M).EQ.1)GO TO 23
      J2=0
      DO 24 N=1,ISLAK2
      IF(ARRAY(M,N).LT.1.E-6)GO TO 24
      J2=1
      IF(J1.EQ.1)GO TO 25
      ICON=M
      J1=1
      GO TO 25
24    CONTINUE
25    IF(J2.EQ.1)GO TO 23
      JND(M)=1
      DO 34 I=1,NSTATE
34    CBASIC(I,M)=0.
      DO 35 I=1,LINES
      II=IBASIC(I)
      IF(II.GT.NSTATE)GO TO 35
      CBASIC(II,M)=ARRAY(I+NUM,IB)
35    CONTINUE
23    CONTINUE
      IF(J1.EQ.0)GO TO 99
      AOPT=1.E-6
      DO 26 N=1,ISLAK2
      IF(ARRAY(ICON,N).LT.AOPT)GO TO 26
      AOPT=ARRAY(ICON,N)
      ICOL=N
26    CONTINUE
      RATIO=1000000.
      DO 29 N=NUM1,NUMCON
      IF(ARRAY(N,ICOL).LT.1.E-6) GO TO 29
      IF(ARRAY(N,IB)/ARRAY(N,ICOL).GE.RATIO)GO TO 29
      IROW=N-NUM
      RATIO=ARRAY(N,IB)/ARRAY(N,ICOL)
29    CONTINUE
      IBASIC(IROW)=ICOL
      NROW=NUM+IROW
      B=ARRAY(NROW,ICOL)
```

```
      DO 30 M=1,IB
30    ARRAY(NROW,M)=ARRAY(NROW,M)/B
      II=NROW-1
      DO 31 I=1,II
      IF(I.GT.NUM)GO TO 62
      IF(JND(I).EQ.1)GO TO 31
62    B=ARRAY(I,ICOL)
      DO 33 M=1,IB
33    ARRAY(I,M)=ARRAY(I,M)-B*ARRAY(NROW,M)
31    CONTINUE
      II=NROW+1
      IF(II.GT.NUMCON)GO TO 28
      DO 32 I=II,NUMCON
      B=ARRAY(I,ICOL)
      DO 32 M=1,IB
32    ARRAY(I,M)=ARRAY(I,M)-B*ARRAY(NROW,M)
28    GO TO 22
99    RETURN
      END
C
      SUBROUTINE COSTPI(P,R,GAMMA,ALFOLD,NSTATE,NTHETA,NALTER,NALPHA,
     1 BETA,IOPT,IDES)
      DIMENSION PI(4),P(4,4,3),R(4,15,3),GAMMA(4,3),ALFOLD(4,60)
      DIMENSION IDES(45),VECTOR(4)
C
C     THIS SUBROUTINE RECEIVES THE INITIAL DATA EACH TIME PERIOD AND
C     RETURNS THE OPTIMAL ALTERNATIVE AND THE MAPPING ONTO THE
C     PREVIOUS PERIOD ALPHA-VECTORS AT THE POINT (1,0,...,0).
C
      PI(1) = 1.
      DO 8 M = 2,NSTATE
8     PI(M) = 0.
      C = 1000000.
      DO 1 I=1,NALTER
      TSUM = 0.
      DO 2 K = 1,NTHETA
      SMIN = 1000000.
      DO 3 M=1,NSTATE
      VECTOR(M) = 0.
      DO 4 N=1,NSTATE
4     VECTOR(M) = VECTOR(M)+PI(N)*P(N,M,I)
3     VECTOR(M) = VECTOR(M)*R(M,K,I)
      DO 5 N=1,NALPHA
      SUM = 0.
      DO 6 M= 1,NSTATE
6     SUM=SUM +VECTOR (M)*ALFOLD(M,N)
      IF(SMIN .LE. SUM) GO TO 5
      SMIN = SUM
      MM=(K-1)*NALTER+I
      IDES(MM) = N
5     CONTINUE
2     TSUM = TSUM + SMIN
      TSUM = BETA*TSUM
      SUM = 0.
      DO 7 N = 1,NSTATE
7     SUM = SUM + PI(N)*GAMMA(N,I)
      SUM = SUM + TSUM
      IF(C .LE. SUM) GO TO 1
      IOPT = I
      C = SUM
1     CONTINUE
      RETURN
      END
```

BIBLIOGRAPHY

1. Bierman, H., and T. Dyckman. Managerial Cost Accounting. New York: The Macmillan Co., 1971.

2. Bierman, H., L.E. Fouraker, and R.K. Jaedicke. "A Use of Probability and Statistics in Performance Evaluation," The Accounting Review, XXXVI (July 1961), 409-17.

3. Bryant, D.J. "Curve Fitting by Markov Transients," Master's Thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, March 1978.

4. Derman, C. "On Optimal Replacement Rules When Changes of State are Markovian," Mathematical Optimization Techniques, (R. Bellman, ed.), University of California Press, 1963.

5. Dopuch, N., J.G. Birnberg, and J. Demski. Cost Accounting--Accounting Data for Management's Decisions, 2nd Ed., Harcourt, Brace, Jovanovich, 1974.

6. Duvall, R.M. "Rules for Investigating Cost Variances," Management Science (June 1967), 631-41.

7. Dyckman, T.R. "The Investigation of Cost Variances," Journal of Accounting Research, Vol. 7 (1969), 215-44.

8. Girshick, M.A., and H. Rubin. "Bayes Approach to a Quality Control Model," Annals of Math. Statistics, XXIII (1952), 114-25.

9. Horngren, C.T. Cost Accounting--A Managerial Emphasis, 4th Ed., Prentice-Hall, 1977.

10. Howard, R.A. Dynamic Programming and Markov Processes, MIT, 1960.

11. Hughes, J.S. "Optimal Timing of Cost Information," Journal of Accounting Research, Vol. 13 (1975), 344-9.

12. Juers, D.A. "Statistical Significance of Accounting Variances," Management Accounting, XLIX (October 1967), 20-5.

13. Kaplan, R.S. "Optimal Investigation Strategies with Imperfect Information," Journal of Accounting Research, Vol. 7 (1969), 32-43.

14. _____. "The Significance and Investigation of Cost Variances: Survey and Extensions," Journal of Accounting Research, Vol. 13 (1975), 311-37.

15. Lee, T.C., G.G. Judge, and A. Zellner. "Maximum Likelihood and Bayesian Estimation of Transition Probabilities," Journal of the American Statistical Association, Vol. 16 (1968), 1162.

16. _____. "Estimating the Parameters of the Markov Probability Model from Aggregate Time Series Data," North Holland: 1970.

17. Luh, F. "Controlled Cost: An Operational Concept and Statistical Approach to Standard Costing," The Accounting Review, XLIII (January 1968), 123-32.

18. Onsi, M. "Quantitative Models for Accounting Control," The Accounting Review, (April 1967), 321-30.

19. Raiffa, H. Decision Analysis--Introductory Lectures on Choices Under Uncertainty, Addison-Wesley, 1968.

20. Roberts, S.W. "A Comparison of Some Control Chart Procedures," Technometrics, 8 (August 1966), 411-30.

21. Rosenfield, D. "Markovian Deterioration with Uncertain Information," Operations Research, 24 (1976), 141-55.

22. Ross, S.M. "Quality Control Under Markovian Deterioration," Management Science, Vol. 17 (May 1971), 587-96.

23. Satia, J.K., and R.E. Lave. "Markovian Decision Processes with Probabilistic Observation of States," Management Science, Vol. 20 (Sept. 1973), 1-13.

24. Shewhart, W.A. The Economic Control of the Quality of Manufactured Profit. New York: The Macmillan Co., 1931.

25. Smallwood, R.D., and E.J. Sondik. "The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon," Operations Research, 21 (1971), 1071-88.

26. Sondik, E.J. "The Optimal Control of Partially Observable Markov Processes," Ph.D. Dissertation, Dept. of Engineering-Economic Systems, Stanford University, June 1971.

27. Sondik, E.J. "The Optimal Control of Partially Observable Markov Processes Over the Infinite Horizon: Discounted Costs," Operations Research, 26 (1978), 282-304.